WebSphere. software

IBM WebSphere Everyplace Connection Manager V5 Handbook

Provide secure wireless access to your e-business applications

Reduce data transmission costs over wireless networks

Deploy mobile access and messaging services

Juan R. Rodriguez
Jean Bidard
Antonio Meneses
JeongRan Lee
Richard Romeo
Pedro Siena Neto
Andreas Rudnik
Paul Starmans
Rhonda Schaffer-Maron
Nestori Syynimaa
Marc Segre

ibm.com/redbooks

Redbooks

International Technical Support Organization

IBM WebSphere Everyplace Connection Manager V5 Handbook

April 2004

SG24-7049-00

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

First Edition (April 2004)

This edition applies to Version 5 of IBM WebSphere Everyplace Connection Manager for multiplatforms AIX, Solaris and Linux.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

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.

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.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

ibm.com®
AIX®
Domino®
DB2 Universal Database™
DB2®
Everyplace®
IBM®
MQSeries®
NavCode®
NetView®
Notes®
PS/2®
Redbooks™
Sametime®
SecureWay®
ThinkPad®
Tivoli®
WebSphere®
Redbooks (logo) ™

The following terms are trademarks of other companies:

Intel, Intel Inside (logos) are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Preface

This IBM® Redbook helps you plan and implement wireless solutions to access backend resources such as databases, application servers, and other legacy applications from wireless devices. This redbook deals with IBM WebSphere® Everyplace® Connection Manager provided functions to enable businesses to make a smooth transition to the wireless Web.

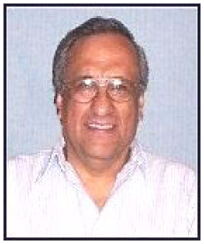
The information provided in this redbook targets Business-to-Employee (B2E) enterprise applications, but most of the scenarios presented apply to Business-to-Consumer (B2C) applications as well. In this redbook you will find examples and scenarios showing ways to extend your enterprise applications to a broad range of mobile devices such as WAP phones, PDAs, and laptops using wireless and dial-up connections. Enterprise applications that can be accessed include WebSphere Application Server applications, portal applications, WebSphere Everyplace Access, MQ Everyplace, Relational Database Synchronization, and others.

In this redbook you will also find sample scenarios showing ways to install and administrate Connection Manager using the Gatekeeper tool to configure support for WAP devices, HTTP services, Mobility Clients, messaging, clustering, roaming, and the available security features such as authentication, data encryption, and digital certificates.

A basic knowledge of wireless and Web technologies is assumed.

The team that wrote this redbook

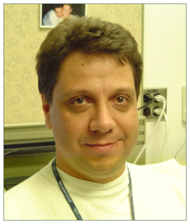
This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.



Juan R.Rodriguez is a Consultant at the IBM ITSO Center, Raleigh. He holds a Master of Science degree in Computer Science from Iowa State University. He writes extensively and teaches IBM classes worldwide on such topics as networking, Web technologies, and information security. Before joining the IBM ITSO, he worked at the IBM laboratory in the Research Triangle Park (North Carolina, USA) as a designer and developer of networking products.

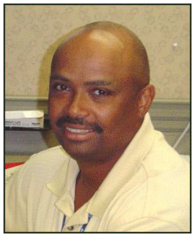
Jean Bidard is an IBM IT Specialist in France based in La Gaude e-Business Solution Center. He has 6 years of experience in integration of IBM and Business Partner products in e-Business and pervasive computing areas. He is working as an Advanced Technical Support for EMEA IGS teams. He demonstrates and designs customer mobile solutions using WebSphere Everyplace Suite components.

Antonio Meneses is a Pre-sales IT Specialist in wireless e-business solutions in IBM Brazil. He works with projects in pervasive, wireless, and telecom solutions, which include the WebSphere Everyplace family of products. He is the leading specialist responsible for wireless e-business projects in Latin America, and is also responsible for the WebSphere Everyplace Connection Manager projects in Latin America.



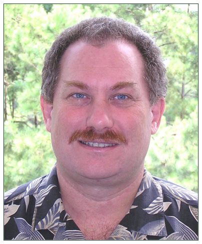
JeongRan Lee is a IT Specialist working for IBM Integrated Technical Services. She joined IBM at 2000 and primarily has been responsible for technical support of the WebSphere product family including WebSphere Application Server and WebSphere Everyplace Server covering areas such as implementing, analyzing, and solving problems of customers' systems with WebSphere products. She holds a bachelor's degree in Computer Science from YonSei University in Seoul, Korea.

Richard Romeo is the Lead IT Wireless Architect with IBM Global Services, Global Web Architecture (GWA) Project Office. He has worked as the lead application architect for building the Global Web Architecture and deployment process, and designing the intranet advance search solution. Most recently he designed a worldwide wireless solution providing IBM mobile employees access to e-mail, calendars, instant messaging, and access to internal sites and applications.



Andreas Rudnik is a Pervasive Solutions Architect for the Pervasive Computing Division in Boeblingen, Germany. He works with the WebSphere family of products on customer projects covering areas such as solution design, architecture review, development of customer-specific product extension, and support in critical situation. His areas of expertise include Pervasive Computing, WebSphere family of products, Java™ 2 Enterprise Edition programming and Solution architectures. He holds a Diplom-Ingenieur degree in Computer Science from Berufsakademie Dresden, Germany.

Ronda Schaffer-Maron is a Wireless Architect with IBM Global Services. Her current assignment involves designing a wireless architecture for the IBM Corporation, which provides worldwide access to mail, calendaring, and other applications through wireless handheld devices. Rhonda has worked for IBM 18 years. Her previous assignments include developing the worldwide replication process for Global Notes® Architecture (GNA) databases, as well as a variety of roles in the Global Web Architecture (GWA) Project Office.



Marc Segre is a Senior Technical Staff Member with IBM's Wireless Solutions Team. He has been a hardware and software developer at IBM for the past 25 years. Marc has designed high-end graphics for IBM's AIX® workstations, then worked in the AIX multimedia group. For the past 10 years, Marc has worked with the financial sector, first on IBM's AIX workstations, then migration to Intel® platforms, and now Wireless Solutions. Marc now focuses on IBM's Websphere platform and security issues related to extending applications to mobile employees and their wireless devices.

Pedro Siena Neto is the Founder and CEO of SST it solutions, an IBM Business Partner in Brazil focused on wireless solutions and applications. He has been working with Pervasive Computing since 1992. He participates on the development of WLAN adapters, and is a pioneer on Logistics Solutions using this technology. Before founding SST, he worked at IBM Brazil, and IBM Research Triangle Park (North Carolina, USA) as a product and development engineer of networking products.

Paul Starmans is an IT Specialist for e-business Enablement Services in IBM Global Services Australia. He joined IBM GSA in 1999, and has spent much of this time working with the WebSphere family of products, primarily Application Server, MQ, and Everyplace Suite, covering areas such as design, implementation, and support. In 2002, he worked on an ITSO residency to produce a course titled *WebSphere MQ Debugging Techniques on Distributed Platforms* in Hursley, UK, and is an IBM Certified MQ Specialist. More recently, his work has been to implement and support a number of WES environments for a commercial customer in Australia. He holds a bachelor's degree of Computer Science from LaTrobe University in Melbourne, Australia.

Nestori Syynimaa is the leading J2EE Architect of a Premium IBM Partner, Sofor Oy, Finland. Syynimaa works in Wireless Solutions research and development. He is responsible for the WebSphere line of products such as WebSphere Application Server and WebSphere Portal. Most of his work is focused on Domino® and J2EE systems integration, and wireless enablement. He is certified as IBM Certified Advanced Application Developer, IBM Certified Advanced System Administrator, IBM Certified Enterprise Developer, MCP (Visual C++), and Java 2.

Thanks to the following people for their contributions to this project:

Ivan Heninger, George Hall, Doug Jones, Henry Welborn, John Kari, Kevin Chu
IBM Research Triangle Park, North Carolina, USA

Glenn E. Miller, Ilyas Guvenilir
IBM USA

Kok Heng Tay
IBM Singapore

Jane Porter, Gill Spencer
IBM United Kingdom

Margaret Ticknor
IBM International Technical Support Organization, North Carolina, USA

Maritza M. Dubec
IBM International Support Organization, San Jose, California

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks™ to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an Internet note to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HZ8 Building 662
  P.O. Box 12195
  Research Triangle Park, NC 27709-2195

1

Introduction

The IBM WebSphere Everyplace Connection Manager (WECM) platform lets you connect mobile computing devices to your company's private intranet and the Internet securely. It uses standards-based protocols to connect over a wide variety of networks, both wireline and wireless, efficiently and easily.

In this chapter we provide a high level overview of this software product including its use within the enterprise, discussion on the benefits, components, and the latest features now available in the new release of IBM WebSphere Everyplace Connection Manager, Version 5.

1.1 Overview

IBM WebSphere Everyplace Connection Manager provides the enterprises the capability to securely expand existing applications to mobile workers over many different wireless networks. Connection Manager creates a mobile Virtual Private Network (VPN) that encrypts data over wireless local area networks (LAN) and wireless wide area networks (WAN) connections. It integrates standard Internet protocols (IP) and non-IP wireless bearer networks, server hardware, device operating systems, and mobile security protocols.

In addition, Connection Manager provides the following advanced functions:

- Seamless roaming between wired and wireless networks
- Enables a secure encrypted tunnel

Figure 1-1 illustrates how these two functions are combined to provide secure and dynamic roaming to access enterprise applications.

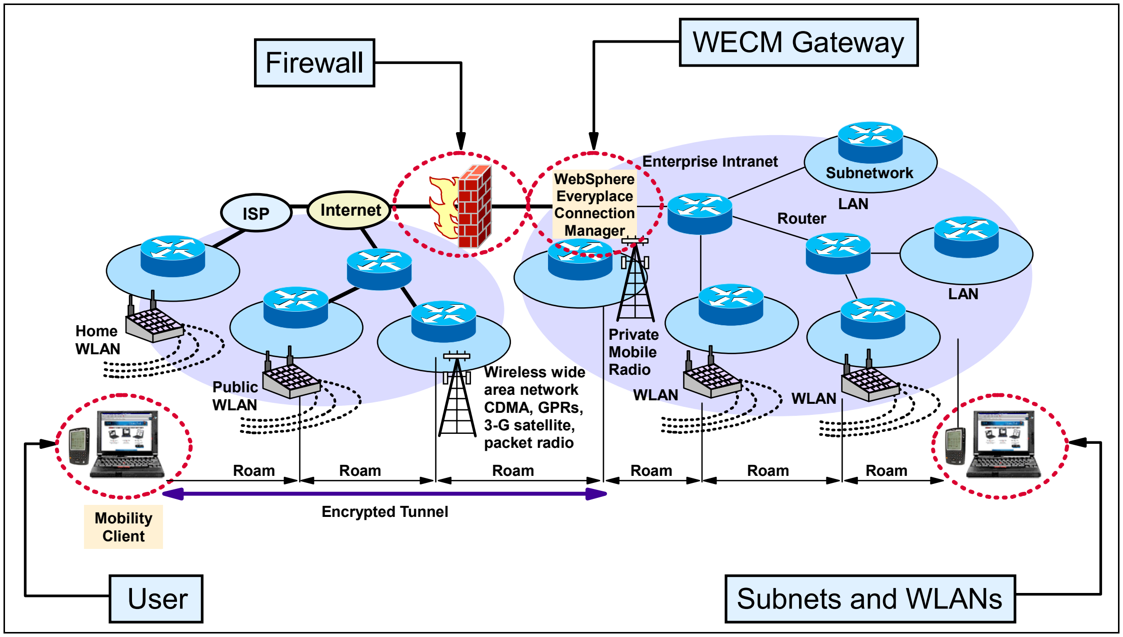


Figure 1-1 The IBM WebSphere Everyplace Connection Manager (WECM) role

Note: In addition to providing virtually seamless roaming between wired and wireless networks, WebSphere Everyplace Connection Manager enables a secure, encrypted tunnel.

1.1.1 Features and functions

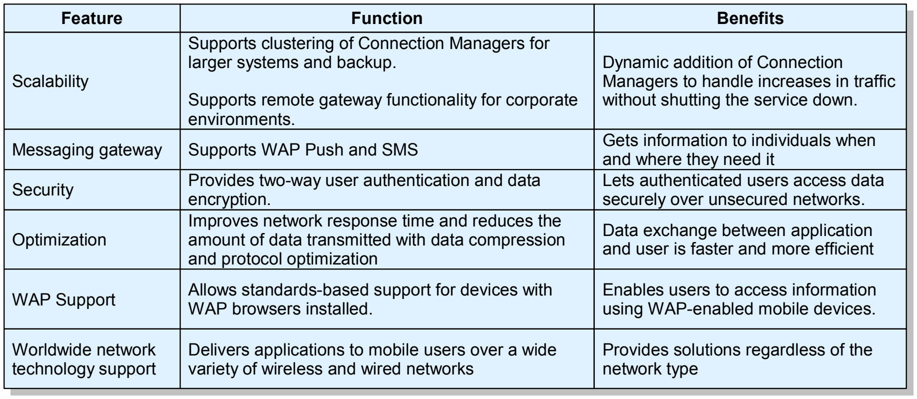
IBM WebSphere Everyplace Connection Manager (WECM) is wireless software and is an open systems communication platform that enables IP applications to run a wireless environment. It gives mobile workers access to host and network resources through radio and dial-up networks.

Connection Manager greatly reduces the cost, complexity, and time required to deploy mobile solution, so enterprise data and applications can be distributed to workers, wherever and whenever they need it. It also extends the corporate network for e-business On-Demand solutions, and protects existing investments in software and information technology infrastructure:

Some of the key functions provided by Connection Manager are:

- Provides enterprise level security with authentication and encryption
- Runs all of your existing TCP/IP applications over wireless networks
- Supports devices that implement the Wireless Application Protocol (WAP) specification
- Supports WAP Push and SMS messaging through the messaging gateway
- Significantly improves wireless network performance while reducing costs through network optimization
- Configurable logging of comprehensive accounting information
- Supports clustering of Connection Manger for larger systems and backup
- Provides a worldwide wireless solution wherever you are through support of international wireless network technologies
- A single WebSphere Everyplace Connection Manager supports any combination of networks (both wired and wireless).

Figure 1-2 illustrates some of the most important key features and benefits.



Feature	Function	Benefits
Scalability	Supports clustering of Connection Managers for larger systems and backup. Supports remote gateway functionality for corporate environments.	Dynamic addition of Connection Managers to handle increases in traffic without shutting the service down.
Messaging gateway	Supports WAP Push and SMS	Gets information to individuals when and where they need it
Security	Provides two-way user authentication and data encryption.	Lets authenticated users access data securely over unsecured networks.
Optimization	Improves network response time and reduces the amount of data transmitted with data compression and protocol optimization	Data exchange between application and user is faster and more efficient
WAP Support	Allows standards-based support for devices with WAP browsers installed.	Enables users to access information using WAP-enabled mobile devices.
Worldwide network technology support	Delivers applications to mobile users over a wide variety of wireless and wired networks	Provides solutions regardless of the network type

Figure 1-2 Connection Manager key features and benefits

New features provided by Connection Manager V5 include the following new functions:

- Roaming enhancements on Windows® clients
- Mobility Client support for Linux, Zaurus
- FIPS 140-2 Certification
- Single Sign-on (SSO) enhancements
- New operating system platforms
- Accounting and billing enhancements
- Add messaging support
- Authentication enhancements
- LDAP architecture updates
- DB2® session management
- User management portlets (administration portlets)
- Installable gateway components and migration
- Enhanced Mobility Client for Palm OS

These new features are described throughout this redbook.

1.2 Connection Manager software

This section provides a high level view of the Connection Manager's wireless environment. Connection Manager provides support for client devices such as Mobility Clients (or Connection Manager clients), PPP clients, HTTP clients, and WAP clients. Figure 1-3 illustrates this environment.

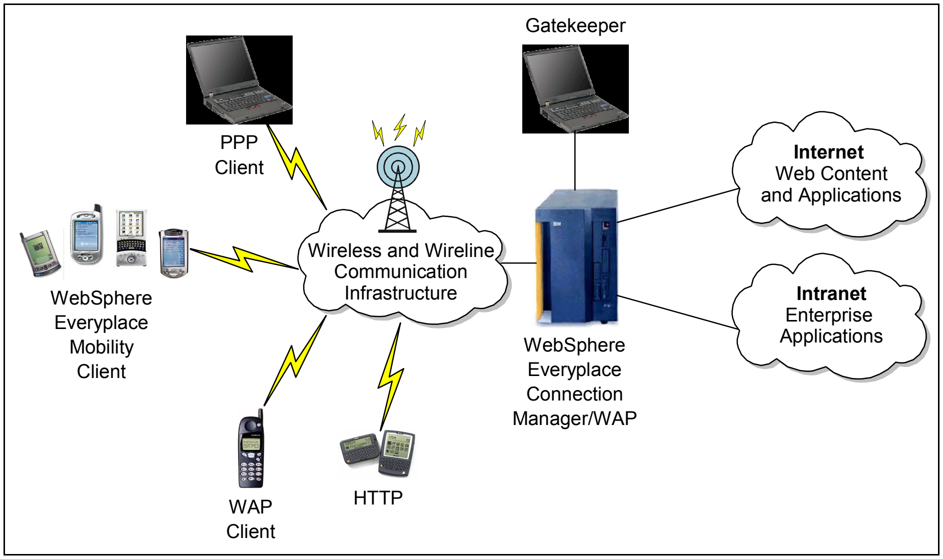

Gatekeeper
PPP Client
Internet Web Content and Applications
WebSphere Everyplace Mobility Client
Wireless and Wireline Communication Infrastructure
Intranet Enterprise Applications
WebSphere Everyplace Connection Manager/WAP
WAP Client
HTTP

Figure 1-3 WebSphere Everyplace Connection Manager environment

Mobility Clients are also called *Connection Manager* clients. The main characteristic of this client is that it installs the Mobility Client software provided by this product. The Mobility Client is software that is installed on a device running the Windows, Palm, and the Linux operating system, and it works in a client-server relationship with Connection Manager to provide secure and optimized IP communications through a wireless network. These clients use the Wireless Link Protocol (WLP) to communicate with the Connection Manager gateway.

The HTTP clients access the Connection Manager gateway using the HTTP or HTTPS protocol. HTTP Access Services in Connection Manager create an optimized and secure tunnel for HTTP communications.

The PPP client is typically a device with its own TCP/IP stack and dialer, and uses a telephone connection to dial the Connection Manager, and uses the gateway as its remote network access server.

The WAP client connects to the gateway, which acts as a proxy for WAP client to access Web-based applications on the Internet or Intranet.

Another type of client (not illustrated in Figure 1-3) is any SMS capable device for which a messaging gateway is provided.

The middle cloud in Figure 1-3 is the wireless network, which provides communications between the clients and the Connection Manager. Following the communication cloud, there is the cloud that represents a wired connection that connects the wireless network to the gateway. The Connection Manager also provides both the WAP gateway proxy and the messaging gateway functions.

Connection Manager administration is another major component of the Connection Manager system. The WebSphere Everyplace Gatekeeper and the administration portlets are used for this function. The Gatekeeper is a Windows, Solaris, and Linux environment application that is used to administer and configure the gateway. For details about the Connection Manager Gatekeeper, see Chapter 8, “Administration” on page 85.

1.3 Connection Manager functions and components

In this section we provide an overview of the main functions and components included in IBM WebSphere Everyplace Connection Manager. These functions and components are illustrated in Figure 1-4.

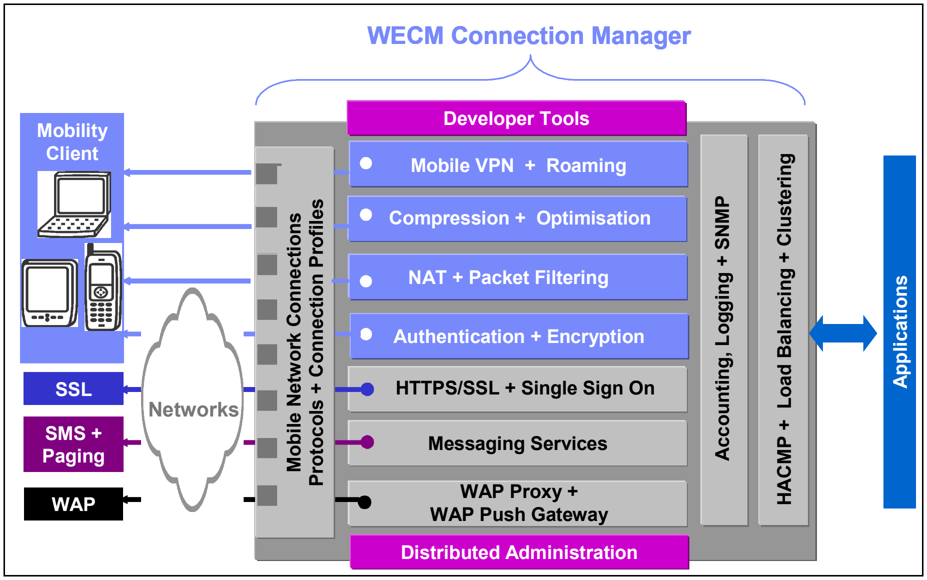

WECM Connection Manager
Mobility Client
SSL
SMS + Paging
WAP
Networks
Mobile Network Connections
Protocols + Connection Profiles
Developer Tools
Mobile VPN + Roaming
Compression + Optimisation
NAT + Packet Filtering
Authentication + Encryption
HTTPS/SSL + Single Sign On
Messaging Services
WAP Proxy + WAP Push Gateway
Distributed Administration
Accounting, Logging + SNMP
HACMP + Load Balancing + Clustering
Applications

Figure 1-4 Connection Manager functions and components

Connection Manager runs on IBM AIX, Sun Solaris, and the Linux operating systems, and provides a standard communications interface to a variety of wireless, dial-up, and LAN networks with data optimization and security. The following functions and components are also included with the product:

- Connection Manager Gatekeeper

  A Java-based administrator's console provides an easy-to-use interface that enables you to configure wireless gateways, define wireless resources, group the resources to control access, and assign administrators to perform operations on the resources as needed. The Gatekeeper enables one or more administrators to work remotely with the Connection Manager. The Gatekeeper provides the capability of defining multiple Gatekeeper administrators, which allows for distributive administrative permissions and responsibilities among them. All administration and configuration data resides in a common Lightweight Directory Access Protocol (LDAP) directory.

- Distributive administration

  Administration of single or multiple Connection Manager machines is simplified with a Java remote console and portlets running on IBM WebSphere Portal Server. This includes defining several levels of administration delegation, each with separate access or change permissions, permitting flexibility to match organizational needs.

- Mobile client

  This provides an optimized and secure IP tunnel for communications with Connection Manager using a variety of wireless and wire line networks.

- Persistent data storage

  This consists of independent databases containing information about the resources comprising your wireless network. The databases are directory services using LDAP and ODBC-compliant relational databases.

- Access Manager program

  On AIX, a Solaris or Linux daemon that manages communications among the Connection Manager Gatekeeper, Connection Manager gateway, and the persistent data storage.

- Mobile Network Interfaces and IP addressing

  WebSphere Everyplace Connection Manager also implements Mobile Network Interfaces (MNI) through which the operating system IP layer communicates with all supported wireless dial or wireline networks. The platform controls one or more IP subnets of users whose traffic is routed through the appropriate MNI. IP addresses can be assigned on either a static basis or through a Dynamic Host Configuration Protocol to support a pool of dynamically assigned addresses.

- Authentication

Connection Manager provides user authentication and data encryption for Mobility Clients using the Wireless Link Protocol (WLP). Also, the Wireless Transport Layer Security (WTLS) for WLP has been implemented in Connection Manager as a specific Mobile Network Connection (MNC), and it can be used in place of the standard Wireless Layer Protocol (WLP) MNC for extra security. In addition, the WAP gateway component provides its own support for secure connections.

- Clustering

  Connection Manager can be configured to be a principle node or subordinate node in a cluster of Connection Managers. In this manner the Connection Manger distributes and services communication request and provides load-balancing efficiency. A cluster manager is automatically installed when you install Connection Manager.

- Security

  Connection Manager provides multiple levels of authentication and encryption to assure the identity of the mobile user, and prevent unauthorized access. Connection Manager incorporates Secure Socket Layer connectivity, Wireless Transport Later Security, and Point-to-Point Protocol (PPP) clients. A symmetric encryption key is used to encode or decode the data with varying key lengths, the strongest being the 256-bit key used in the Advance Encryption Standard algorithm. To provide data privacy and protection, the Connection Manager security component provides an extensive cryptographic library, which includes Data Encryption Standard (DES), Triple DES, and RC5, SEED or Advance Encryption Standard (AES) algorithms.

- HTTP/HTTPS Access Service

  For mobile devices with an SSL-capable browser, Connection Manger provides support for either unsecured Hypertext Transfer Protocol (HTTP) or authenticated Hypertext Transfer Protocol over Secure Socket layer (HTTPS) connectivity. To enable Single Sign-On through Web servers or portals, WebSphere Everyplace Connection Manager Version V5.0 adds Light Weight Third Party Authentication token support.

- Mobility Client

  The Mobility Client software runs locally on your mobile device, and provides a full-function interface to communicate with Connection Manager. Upon authenticating to the Connection Manager, a VPN is established and the device securely joins the enterprise intranet. The Connection Manager supports standard IP routing even over non-IP wireless bearer networks to ensure unbroken, end-to-end TCP sessions between mobile devices and application servers. The Mobility Client includes a toolkit and application program interfaces (APIs) to create network-aware applications.

- Mobile Access services

A Mobile Access service provides an encrypted tunnel securing a wireless connection between the Connection Manager and the Mobility Client.

- Messaging services

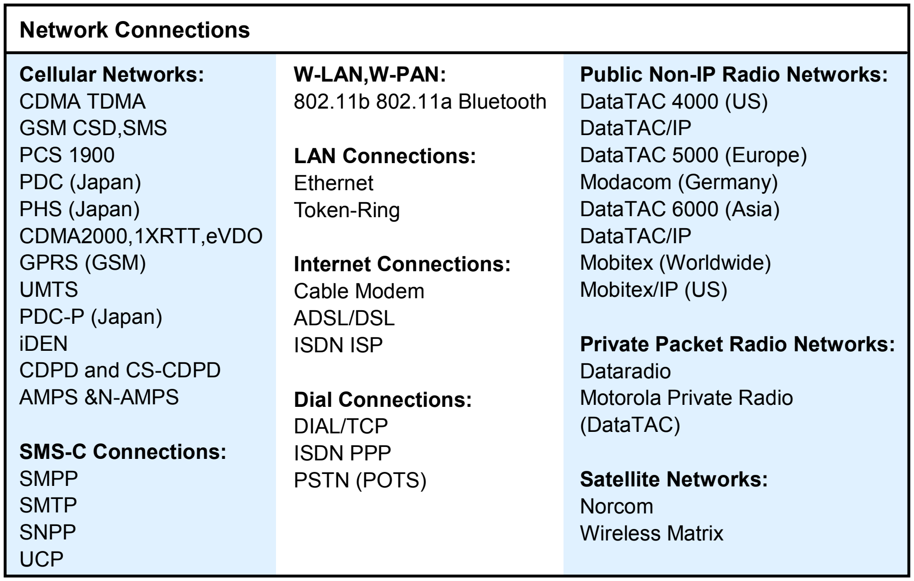
  Connection Manager messaging services support several types of messaging modes including Short Message Services (SMS), e-mail using Simple Mail Transfer Protocol (SMTP), unconfirmed WAP push, message delivery over proprietary networks such as Mobitex or Motient, and Simple Network Paging Protocol (SNPP). Additionally, Connection Manager includes the Wireless Communication Transfer Protocol (WCTP) delivering wireless messages, both one and two-way to appropriate receiving devices such as pagers, mobile phones, or other wireless devices.

- Connectivity

  Connection Manager supports a variety of wireless and dial-up network technologies. All traffic flows through the Connection Manager uses a Mobile Network Connection (MNC). A MNC is a resource that is assigned to a Connection Manager and defines a specific type of network connection.

  The MNC consists of a line driver, a network protocol interpreter, and one or more physical ports. You configure at least one MNC for each network provider that you will use. There is a different MNC for each specific type of network or bearer through which mobility or WAP clients connect.

  The supported networks are shown in Figure 1-5.



Network Connections

Cellular Networks:
CDMA TDMA
GSM CSD,SMS
PCS 1900
PDC (Japan)
PHS (Japan)
CDMA2000,1XRTT,eVDO
GPRS (GSM)
UMTS
PDC-P (Japan)
iDEN
CDPD and CS-CDPD
AMPS &N-AMPS

SMS-C Connections:
SMPP
SMTP
SNPP
UCP

W-LAN,W-PAN:
802.11b 802.11a Bluetooth

LAN Connections:
Ethernet
Token-Ring

Internet Connections:
Cable Modem
ADSL/DSL
ISDN ISP

Dial Connections:
DIAL/TCP
ISDN PPP
PSTN (POTS)

Public Non-IP Radio Networks:
DataTAC 4000 (US)
DataTAC/IP
DataTAC 5000 (Europe)
Modacom (Germany)
DataTAC 6000 (Asia)
DataTAC/IP
Mobitex (Worldwide)
Mobitex/IP (US)

Private Packet Radio Networks:
Dataradio
Motorola Private Radio
(DataTAC)

Satellite Networks:
Norcom
Wireless Matrix

Figure 1-5 Supported network connections

- WAP proxy and Push Proxy Gateway (PPG)

  Connection Manager can be configured as a WAP proxy and provides connectivity for WAP V1.1 and WAP V1.2 client services. Connection Manager fully supports the WAP Wireless Session Protocol (WSP), which links the micro browser with cell phones and PDAs. The Connection Manager WAP Push Proxy Gateway function allows external applications to push various content down to a WAP device. Wireless Transport Layer Security (WTLS) secures the connection from the Connection Manager to the WAP client device. The Secure Sockets Layer is used to establish a secure connection from the Connection Manager to Web backend servers.

  Connection Manager allows standard-based support for devices with WAP browsers installed. For information about the WAP gateway (proxy) provided by Connection Manager, see Chapter 19, “WAP gateway” on page 415.

1.4 Connection Manager component interaction

Connection Manager can be configured in a “stand-alone” configuration, but in many cases it can also be configured to interact with a WebSphere Everyplace Suite (WES) configuration. Figure 1-6 illustrates such a configuration.

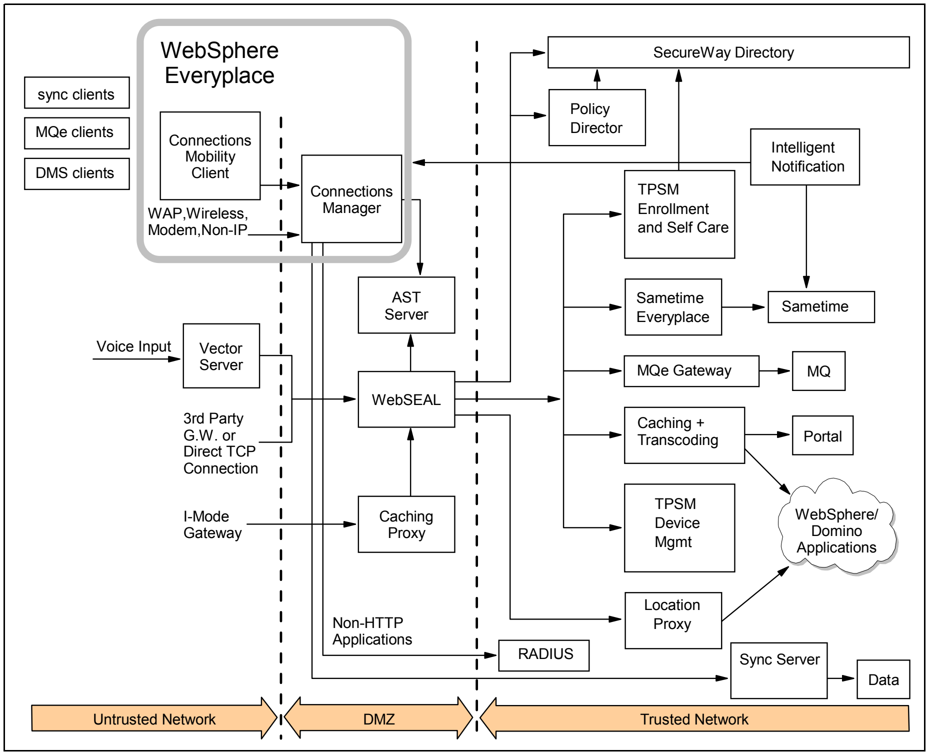


Figure 1-6 Connection Manager component interaction

The Connection Manager is the entry point for all requests coming from Mobility Clients. When a new request arrives at the Connection Manager, it is passed to an authentication component (WebSEAL Lite in this example). WebSEAL Lite is a plug-in that is installed with Web Traffic Express, or WTE. WTE can also serve as the HTTP caching proxy required for WAP support. It is also known as the “Edge server caching proxy.”

Lower on the diagram, the box labeled *Radius* is an optional component that is also used for authentication of clients. If the Radius server is available, the authentication proxy will work with the Radius server to determine if a client is authorized to use the system. After a client has been authorized, it is passed on to access whatever Web site it is trying to access (as shown by the cloud labeled *Internet* in the diagram).

Whenever a WAP client requests HTML information from the Internet, it must be converted to WML before being passed back to the WAP client. That function is handled by the box labeled *Caching & Transcoding*. Its purpose is to transform HTML responses to WML responses. In addition to this transformation, the transcoding component has other capabilities such as removing large graphics. Although if the client is a PDA, the transcoding component will provide more information (such as graphics) based on the information provided in the response headers that define the type of device requesting the data.

The box labeled *SecureWay Directory* is the LDAP server that all of Connection Manager's various components use to keep track of configuration information.

1.5 TCP traffic optimization

Connection Manager gateway and the Mobility Client (running the Mobility Client software) provide traffic optimization by incorporating the following function in the Wireless Link Protocol (WLP):

- TCP-Lite
- HTTP Codec
- Compression
- Packet joining
- Packet fragmentation
- TCP retransmission suppression

Figure 1-7 illustrates the traffic optimization functions available for Mobility Clients.

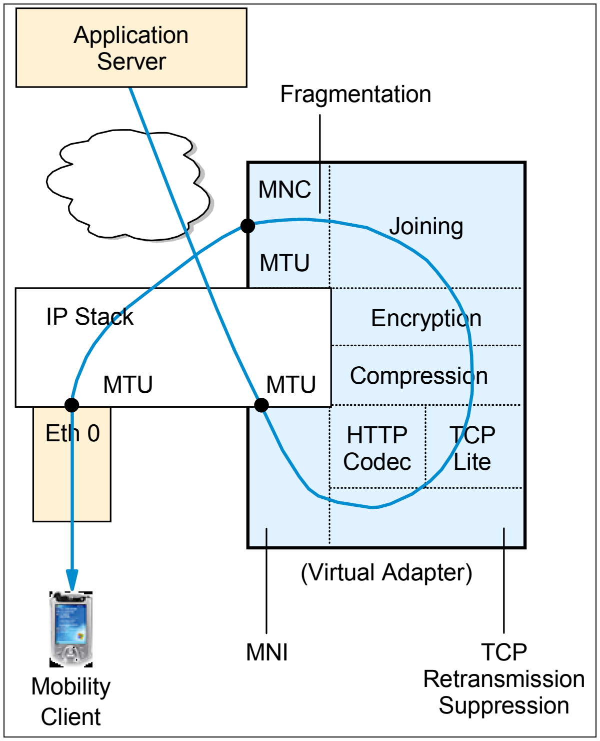


Figure 1-7 TCP traffic optimization

TCP-Lite

TCP-Lite is a service that reduces the amount of the data transferred between Mobility Client and Connection Manager gateway. This is done by dropping unnecessary TCP protocol data units (PDUs) that are related to session management rather than to the data transfer.

HTTP Codec

HTTP Codec is a service that provides reduction of the HTTP data stream by removing and encoding header fields of the stream. HTTP Codec uses TCP-Lite as an underlying protocol.

Compression

Compression is a service that compresses the actual data of the traffic that is transferred between the Mobility Client and the Connection Manager gateway.

Packet joining

Packet joining is a service that joins multiple small message transfer units (MTUs) into a bigger one before transferring the MTU to the underlying network. This reduces the amount of the transferred MTUs by optimizing the size of the MTU and by dropping the unnecessary duplicate header bytes.

Packet fragmentation

The packet fragmentation service fragments big MTUs into smaller ones to optimize the use of MTUs before transferring the MUTs to the underlying network. This will save you time because in general it is more efficient to do fragmentation in the application layer rather than let the network layer do it.

TCP retransmission suppression

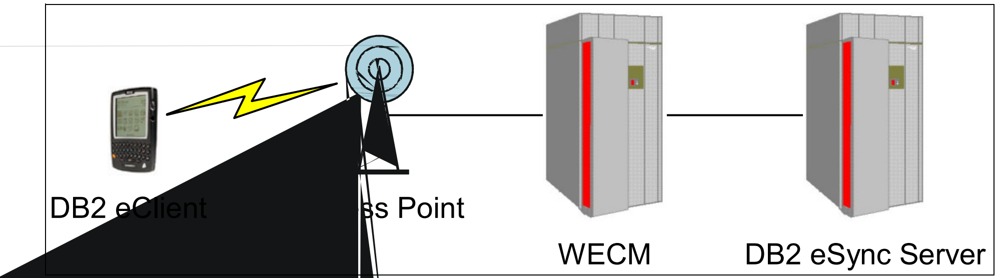
TCP retransmission suppression service reduces the amount of unnecessary TCP retransmission packets, which may occur especially in slow networks. The service holds the packets for a specified amount of time, and if the retransmission of the packet occurs within that time, it drops the retransmitted packet rather than sending it again to the network because the acknowledge for the original message can already be on its way.

1.6 Mobility applications

In this section we include some of the most common applications using the Mobility Client and the HTTP Access Services provided by Connection Manager.

1.6.1 DB2e applications with Connection Manager

DB2 Everyplace applications connect to the Everyplace Connection Manager for data synchronization and other functions. The Everyplace Mobility Client must be installed in the client device. Figure 1-8 illustrates the connection for this scenario.



plications with Connection Manager

er

In addition to the function provided by DB2 Everyplace, the following features are provided and can be enabled in the Everyplace Mobility Client to improve the wireless connection:

- **Data compression**. Mobility Client capability to compress data can reduce the amount of transferred bytes up to four times the usual traffic. This is quite useful if wireless network provider billing is based on transferred bytes as GPRS network providers usually do. Synchronization is also faster, because the number of the over-the-air packages is smaller.
- **Security**. Mobility Client encrypts all traffic, so you can securely access your corporate DB2 Everyplace applications even when using public networks. This way you can synchronize your databases safely without any concerns about the privacy of the transferred information.
- **Roaming**. Roaming capabilities of Mobility Client always give you an optimized network connection when multiple networks are available. You can set the cost of the network connection based on price or speed of the connection, whatever is better to your application. Mobility Client automatically changes to the best available network based on your configurations. For example, in the office you may have the Bluetooth connection in a warehouse, the WiFi connection, and in the car the GPRS connection. Mobility Client changes used networks “on the fly,” and your DB2e application can synchronize data wherever you are.

1.6.2 MQ Everyplace applications

MQ 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 through their mobile devices.

Note: MQSeries Everyplace consists of Java and C components enabling solution developers to create an MQSeries Everyplace gateway and client on a variety of devices and platforms.

MQSeries Everyplace can function independently using the facilities of the Connection Manager only to support wireless communications, as it has its own security management and transport protocol. MQSeries Everyplace can also use the facilities of Connection Manager to communicate with a servlet, which incorporates the MQe class library. In this HTTP mode, an application can overcome issues of firewall configuration (for example, port selection), and authenticate itself through the implemented mechanisms by placing the user ID and password in the HTTP header. MQSeries Everyplace components are illustrated in Figure 1-9.

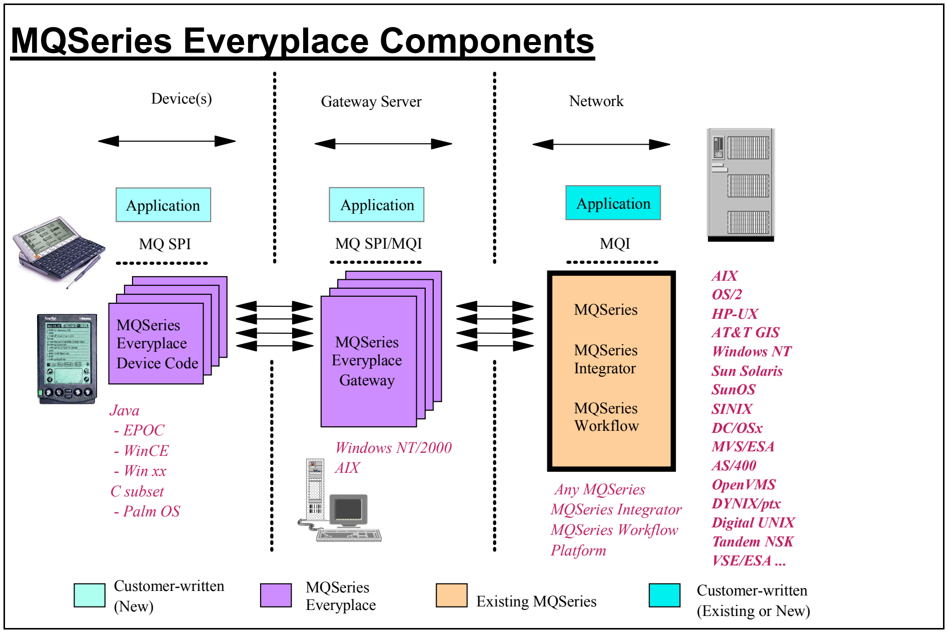


Figure 1-9 MQSeries Everyplace components

A MQSeries Everyplace application on the server side runs in a JVM and listens on a specific port. Only one Queue Manager can run in a given JVM. In order to scale the application, several JVMs must be started, each listening on a different port. The client-side application must be configured to address one of the ports. In other words, load balancing must be managed by the user.

In addition, the purpose of MQSeries Everyplace is to provide a once-only assured delivery of messages for applications running on devices with one or more of the following characteristics:

- The device typically cannot support a fully configured MQSeries queue manager.
- The device connects using a wireless protocol.

The type of devices that use MQSeries Everyplace are:

- Personal Digital Assistants (PDAs)
- Phones

- Sensors
- Laptops

As these sorts of devices are typically used outside of an organization’s intranet, security is an important factor. MQSeries Everyplace provides comprehensive security capabilities to address this potential problem.

MQSeries Everyplace provides assured messaging capability over non traditional networks such as those now available for wireless connections. The connectivity support provided by Connection Manager to support Mobility Clients means that applications that use MQSeries Everyplace can use this connection support to allow them to run on wireless devices.

One of the advantages of using Connection Manager is that it can be configured to provide authentication and encryption services. Enabling authentication means that when an end user establishes a wireless connection, they are prompted for their authentication details by the wireless client. Enabling encryption means that all data transferred between the client and the server is encrypted, preventing unauthorized people from viewing the data.

These authentication and encryption services of Connection Manager can be of use in addition to any authentication and encryption that applications or other products may use communicating through the Connection Manager.

For information about MQSeries Everyplace in a Connection Manager environment, see Chapter 13, “MQe application traffic optimization” on page 247.

1.6.3 WebSphere Everyplace Access applications

As the popularity of mobile computing grows, mobile workers are realizing the value of having access to information and resources held by the enterprise. Access to this information, and the enterprise resources allows them to work more effectively, and stay up to date with changing events and new information while away from the office. Likewise, enterprises are seeing real benefit from providing their mobile work force with mobile devices, and giving access to enterprise information and applications. With advances in WebSphere Everyplace Access, the enterprise can also push important information to the mobile worker.

WebSphere Everyplace Access is a comprehensive product that provides end-to-end coverage for the enterprise mobile computing needs. Everyplace Access is middleware that enables the enterprise and business partners to create robust mobile computing solutions that extend enterprise resources such as business applications, business data, and business information to the mobile worker. Everyplace Access consists of both the *infrastructure* intended to reduce the complexity of providing mobile computing solutions, and the *services* needed to create the right mobile solution for the mobile worker and their particular needs.

Everyplace Access provides the connection between the mobile client and the enterprise environment. Figure 1-10 provides the logical view of the overall mobile computing runtime environment. The figure shows the following major areas.

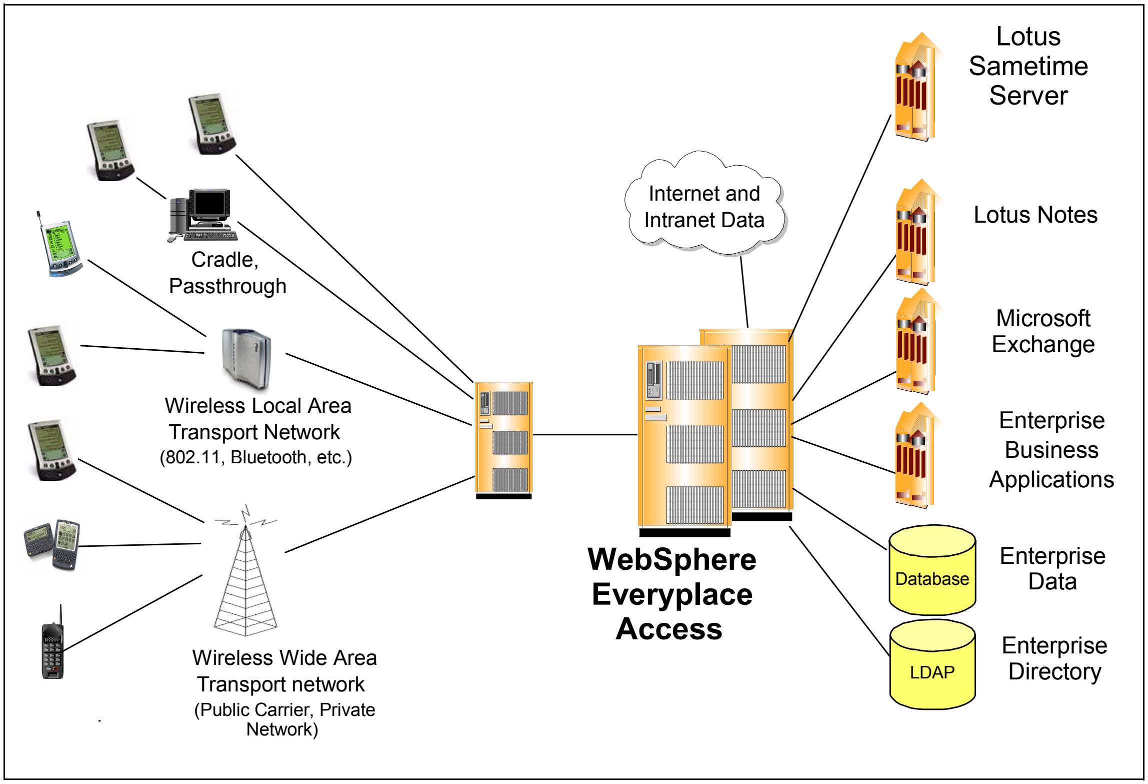

Lotus Sametime Server
Lotus Notes
Internet and Intranet Data
Cradle, Passthrough
Microsoft Exchange
Wireless Local Area Transport Network (802.11, Bluetooth, etc.)
Enterprise Business Applications
WebSphere Everyplace Access
Database
Enterprise Data
LDAP
Enterprise Directory
Wireless Wide Area Transport network (Public Carrier, Private Network)

Figure 1-10 Logical view of mobile computing with WebSphere Everyplace Access

By being the access point to the enterprise, Everyplace Access addresses the complexity of mobile computing within the Enterprise environment, and simplifies the overall runtime environment. The logical view in Figure 1-10 shows Everyplace Access in support of the enterprise and identifies it as providing the consistent access point for the mobile users regardless of the mobile device, the network, or the type of request or interaction with the enterprise that is needed.

Everyplace Access follows the IBM software development ground rules of incorporating open standards-based technology into the product and reusing existing technology wherever appropriate. Both IBM’s WebSphere Application Server and WebSphere Portal technologies provide the foundation for Everyplace Access. These products are built using open standards and open technology such as XML and Java. Mobile computing technologies are changing fast and quickly evolving, which makes Everyplace Access the stabilizing factor in a very dynamic world. Everyplace Access accomplishes this by allowing the solution developer to write to its services, which minimizes the need to understand the details of the underlying technologies.

For a sample integration of Connection Manager with WebSphere Everyplace Access, see Chapter 15.1, “WebSphere Everyplace Access integration” on page 304.

2

Installation planning

This chapter discusses the issues related to planning a WebSphere Everyplace Connection Manager (WECM) installation. Given the broad range of platforms and topologies available for a Connection Manager implementation, this chapter will help address the ways in which the product may be used, and how to plan for a successful installation.

In this chapter, the following topics are discussed:

- Hardware and software requirements
- Implementation topologies
- Installation tasks
- WebSphere Everyplace Suite considerations

2.1 Connection Manager product distribution

WebSphere Everyplace Connection Manager V5 code base is currently available in three different bundled packages:

- Starter Edition:
  - Contains WebSphere Everyplace Connection Manager runtime, mobile access services, messaging services, and the WAP proxy
  - Limited by license to 50 users
  - Useful for use in a proof of concept, pilot, or small production setup
- No WAP
  - Contains WebSphere Everyplace Connection Manager runtime, mobile access services, and messaging services
  - For installations that do not require WAP services
- Full
  - Contains WebSphere Everyplace Connection Manager runtime, mobile access services, messaging services, and the WAP proxy
  - For large environments that require access to WAP services

2.2 Supported operating systems

WebSphere Everyplace Connection Manager V5 is supported by the following operating systems:

- IBM AIX V5.1 and V5.2
- Sun Solaris V7, V8, and Trusted Solaris 8
- Linux:
  - Red Hat 7.3 and 8.0
  - SuSE 7.3, 8.0, and 8.1
  - UnitedLinux 1.0

2.2.1 Gatekeeper

Connection Manager V5 Gatekeeper software is supported on the following operating systems with the TCP/IP protocol installed:

- IBM AIX V5.1 or later
- Sun Solaris V7, V8, and Trusted Solaris 8
- Microsoft® Windows:
  - 98
  - NT

  - 2000
  - Me
  - XP
- Linux
  - Red Hat 7.2
  - SuSE 7.2
  - TurboLinux Workstation 7.0

2.2.2 Mobility Clients

WebSphere Everyplace Connection Manager V5 Mobility Clients are supported on the following operating systems with the TCP/IP protocol installed:

- Palm OS 3.5.x, 4.x
- Microsoft Windows
  - 98
  - NT
  - 2000
  - Me
  - XP
- Microsoft Windows CE V3.0 on a handheld PC 2000 device or a Pocket PC 2002 device on the Pocket PC platform
- Sharp Zaurus SL5600 using embedded Linux
- Linux desktop
  - Red Hat 7.3, 8.0
  - SuSE 7.3, 8.0, 8.1
  - UnitedLinux 1.0

For details on installing and configuring Mobility Clients, refer to Chapter 10., “Mobility Clients” on page 171.

2.3 Storage requirements

Prior to beginning the installation, consideration should be made as to the amount of storage required by WebSphere Everyplace Connection Manager. This can be divided into two key areas:

- Virtual storage
  - This is the sum of the RAM plus the paging space available on the physical disk.

- Disk storage
  - Physical disk storage will be used both for installation of the products and during runtime.

2.3.1 Virtual storage

Around 15 MB of virtual storage is required for the first Gatekeeper logged into the Access Manager, and approximately 5 MB for each additional Gatekeeper.

The Connection Manager will use approximately 35 MB of virtual storage, regardless of the number of connections. An additional 35 KB is required for each connected user or mobile device.

On a UNIX® host, run the **`vmstat`** command to check the amount of available virtual storage.

2.3.2 Disk storage

For each packet transmitted, a 24-byte accounting record may be generated. Assuming an average account produces 500 packets per day, this would require 12 KB of disk space. Of course, this will vary depending on the number of users, and the traffic pattern and load through the Connection Manager.

Consideration must be given to the amount of logging required. When full logging is used on the Connection Manager, many megabytes of logged data may be produced within a few minutes. To reduce the amount of disk storage required for logging, a configuration that logs only errors and warnings is recommended. In a stable environment, this may only produce a few kilobytes of data over a week long period.

By default, logs are written to /var. Run the **`df -k /var`** command to ensure there is sufficient free space available for your logging requirements.

2.4 Network considerations

To help simplify the setup and configuration of a WebSphere Everyplace Connection Manager environment, the way in which the network will be used should be closely considered prior to beginning the installation. Areas such as IP address ranges, port usage, firewall configuration, traffic routing, and the type of network providers your Connection Managers will use all need to be looked at.

In addition, an understanding of the type of load that will be placed on the network is required to ensure the environment is built in such a way so as to perform to expectations. This will help to determine whether multiple Connection Managers may be required to service the load, and the way in which to configure the various components.

2.4.1 Infrastructure requirements

Within the enterprise network, a range of IP addresses need to be made available to WebSphere Everyplace Connection Manager. Specifically, you need to decide what subnet will be used to define an MNI to a mobile access services, and how many subnets are required. These subnets must be large enough to cater for the required number of concurrently connected users, and must be unused within the enterprise network. In the case where NAT is used, these requirements will alter as the range of addresses supplied by WebSphere Everyplace Connection Manager can be greatly increased beyond that of the range available in the enterprise.

Once the addressing details have been confirmed, ensure that each of the hosts within the environment is able to route to the new subnets. This will usually involve modifying the routing tables to add static routes for routing traffic back to a defined MNI. In a single server installation, this will not be an issue, however, where multiple hosts are involved, routing does become more complex. Unless additional hosts have a defined route to the MNI, they will not be able to route packets back to the clients.

2.4.2 Use of ports

WebSphere Everyplace Connection Manager requires an number of different TCP ports to be available. The ports required will depend on the Connection Managers configuration and the services required by the clients. Depending on your network infrastructure, configuration changes may need to be made to any firewalls to allow traffic on these ports.

Figure 2-1 on page 26 shows a sample topology of a WebSphere Everyplace Connection Manager installation with two firewalls. The firewalls are used to block all unwanted connections from the Internet as well as from the Intranet. Only known connections to and from WebSphere Everyplace Connection Manager are enabled in the firewalls.

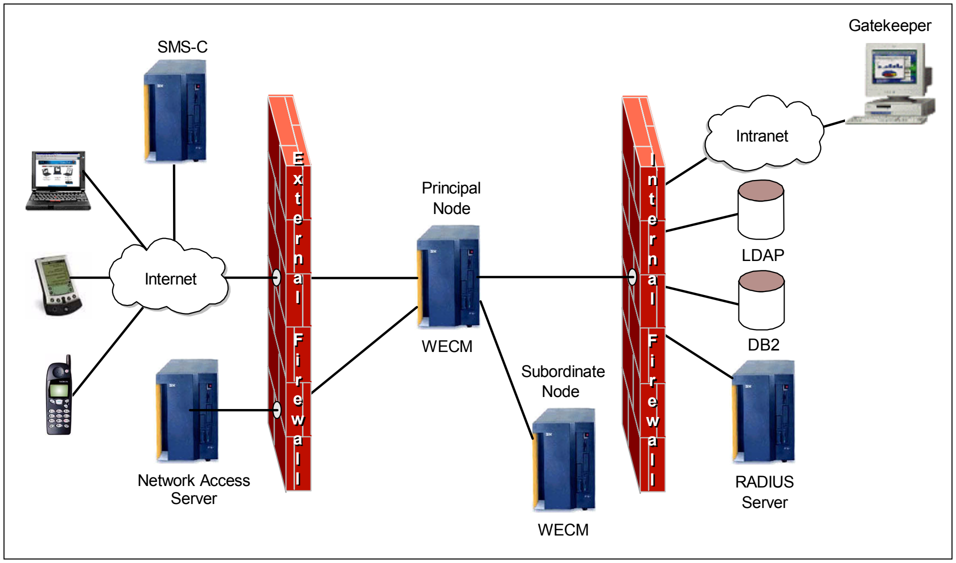


Figure 2-1 Firewalls

The external firewall is used to block all unwanted connections from the Internet. Only connections to the mobile access service, WAP proxy, HTTP access service, and the messaging service are allowed. See Table 2-1 for detailed port information that must be enabled in the external firewall.

All ports may be modified by using the Gatekeeper to change the relevant port number. Only those relating to the configured services will be required.

Table 2-1 External firewall

Port	Direction	Source address	Destination address	Service description
443/tcp	In	any	Connection Manager external IP address	HTTP access service
1812/udp or 1645/udp	In	NAS	Connection Manager external IP address	RADIUS authentication messages
1813/udp or 1646/udp	In	NAS	Connection Manager external IP address	RADIUS accounting messages

Port	Direction	Source address	Destination address	Service description
8888/udp	Both	any or Connection Manager external IP address	Connection Manager external IP address or any	Change password
8889/udp	Both	any or Connection Manager external IP address	Connection Manager external IP address	IP-LAN receive
9200/udp	In	any	Connection Manager external IP address	Connection-less WAP service
9201/udp	In	any	Connection Manager external IP address	Connection-oriented WAP service
9202/udp	In	any	Connection Manager external IP address	Secure connection-less WAP service
9203/udp	In	any	Connection Manager external IP address	Secure connection-oriented WAP service
9610/udp	In	any	Connection Manager external IP address	RADIUS authentication messages from Mobility Clients
13131/tcp	In	SMC-C	Connection Manager external IP address	Messaging services
13132	In	SMC-C	Connection Manager external IP address	Secure messaging services

The internal firewall separates the Connection Manager from all other systems (such as DB2, LDAP, etc.) and the Intranet. Table 1-2 lists all ports that need to be enabled in the internal firewall.

Table 2-2 Internal firewall

Port	Direction	Source address	Destination address	Service description
389/tcp	Out	Connection Manager internal IP address	Remote LDAP server	Connection Manager access to remote LDAP server
1812/udp or 1645/udp	Out	Connection Manager internal IP address	RADIUS server	RADIUS authentication messages
1812/udp or 1646/udp	Out	Connection Manager internal IP address	RADIUS server	RADIUS accounting messages
9555/tcp	In	Gatekeeper	Connection Manager internal IP address	Gatekeeper administration
9559/tcp	In	Gatekeeper	Connection Manager internal IP address	Gatekeeper administration with SSL
13131/tcp	In	any	Connection Manager internal IP address	Messaging services
13132/tcp	In	any	Connection Manager internal IP address	Secure messaging services
14356/tcp	Out	Connection Manager principal node IP address	Connection Manager subordinate node IP address	Subordinate node listens for requests from principal node

Port	Direction	Source address	Destination address	Service description
50000/tcp	Out	Connection Manager internal IP address	Remote DB2 server	Connection Manager access to remote DB2 server
/	In	Connection Manager VPN IP address pool	any	Any type of application that is used by the Mobility Client

All ports may be modified by using the Gatekeeper to change the relevant port number. Only those relating to the configured services will be required.

Note: On Solaris systems, port 8888 is commonly used by AnswerBook2. Verify the availability of this port prior to installation using the **`netstat`** command.

2.5 Installation overview

The following section describes the ways in which a WebSphere Everyplace Connection Manager environment may be configured, and the steps involved in creating such environments. This is intended as a high level guide providing examples of network topologies and steps involved in such a configuration. Thorough planning prior to installation will help ensure that the environment is able to support the predicted loads, while also being easily scalable as usage requirements grow and the load increases.

2.5.1 Sample topologies

The following chapter aims to identify some examples of the various topologies that may be used when creating a WebSphere Everyplace Connection Manager environment. The size of the environment will vary according to the number of users required to access the environment. Figure 2-2 shows four different scenarios on how WebSphere Everyplace Connection Manager can be set up.

While the servers may be a mix of operating systems, each is assumed to meet the minimum hardware and software requirements. The other range of products shown for use in the samples with WebSphere Everyplace Connection Manager may be any of those from the list of supported products. While the sample topologies are not intended to indicate the only method of configuring

WebSphere Everyplace Connection Manager and related software, it aims to provide a planning reference from which to use and build upon.

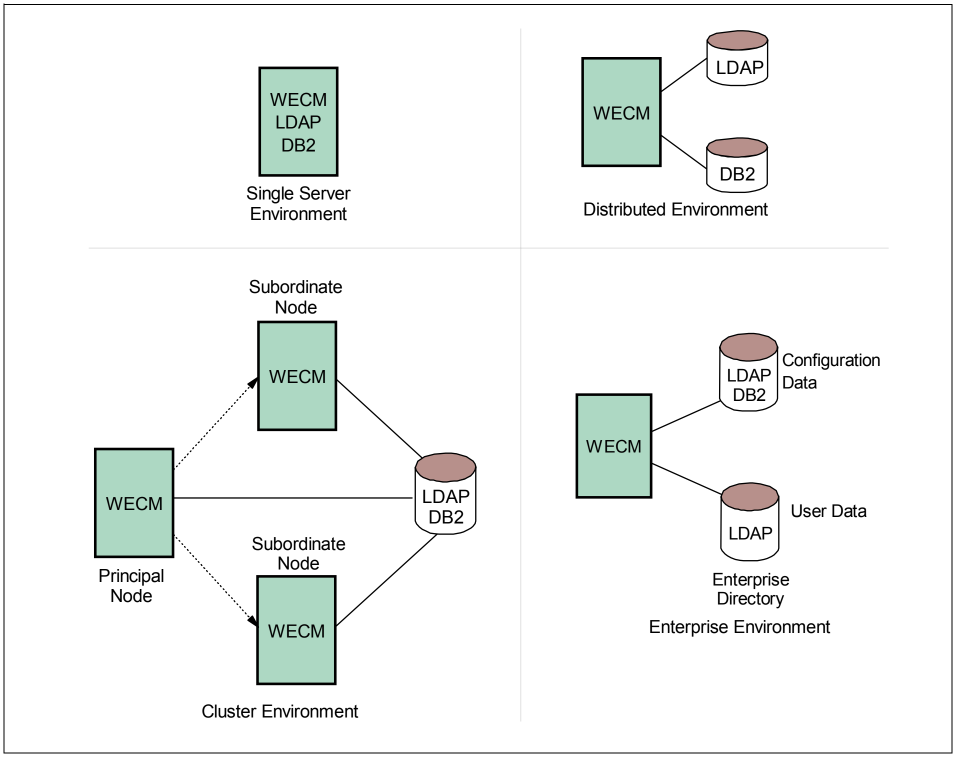


Figure 2-2 WebSphere Everyplace Connection Manager (WECM) topologies

The different scenarios can also be combined. The section titled “ISP environment” on page 31 gives an example of what a combination of a distributed, clustered multi-enterprise environment can look like.

Single-server environment

Hosting all components on a single server is a simple and cost effective way of creating a proof of concept environment. While overall performance may be reduced due to the high resource usage, it is an ideal way to demonstrate and investigate WebSphere Everyplace Connection Manager configurations.

If your Web server supports it, it may also be configured to perform the role of the proxy server, thus removing the need for a separate proxy server product. Data to be accessed by the clients may also reside on this host, creating a completely self contained environment.

Distributed environment

Here the components are shared across three separate hosts. This reduces the load on an individual server, thereby increasing performance as compared to the single-server scenario. By separating WebSphere Everyplace Connection Manager from the database and LDAP server, this removes the situation where WebSphere Everyplace Connection Manager and the database are competing for system resources. Provided the network link between WebSphere Everyplace Connection Manager and the database is reliable and fast, this configuration will benefit from a good gain in performance.

Using multiple servers in an environment also allows tighter security control through the use of firewalls and the creation of a tiered architecture.

Cluster environment

Another technique used for effectively sharing load and improving performance is Connection Manager clustering. This is useful in larger scale environments as discussed in Chapter 22., “Clustering” on page 473. Once WebSphere Everyplace Connection Manager server is set up as the principal node, all other systems are running as subordinate nodes. The principal node will distribute incoming requests to the subordinate nodes based on a distribution algorithm.

Enterprise environment

WebSphere Everyplace Connection Manager provides the option to separate the data store for configuration data and user data. In this way, Connection Manager can use an existing LDAP to authenticate users. The user data can also be distributed over multiple LDAP servers.

ISP environment

The ISP environment is a combination of the previous topologies. It uses the cluster scenario to split the environment into several independent segments. The ISP runs the principal node with all configuration data. Each enterprise has its own infrastructure of WebSphere Everyplace Connection Manager subordinate nodes and LDAP servers that provide the user data. Further information can be found in Chapter 22, “Clustering” on page 473.

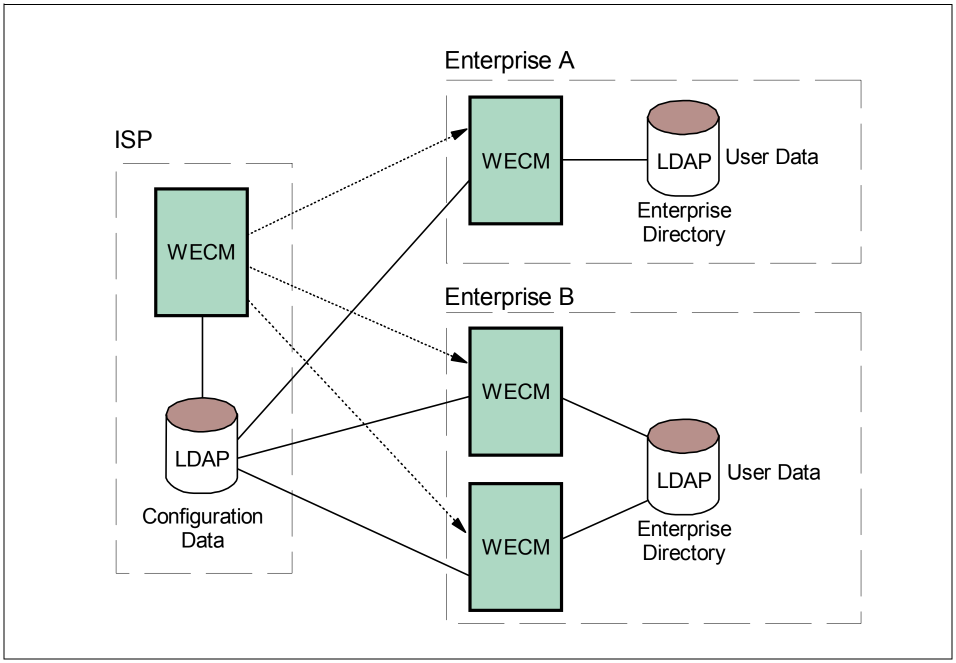


Figure 2-3 ISP environment

2.5.2 Installation process

The following section describes a high level overview of the installation process. This will help to plan the tasks involved in setting up the WebSphere Everyplace Connection Manager environment:

1. Prepare and configure the servers:
   - Install and configure operating system.
   - Ensure appropriate operating system patches have been applied.
   - Configure the networking.
2. Install and configure Web server:
   - This will be used for DSS administration as well as for serving data if required.
3. Install the database server.
4. Install and configure DSS.
   - Configure the administrator's user ID and password.
   - Prepare the DSS database.
   - Configure the Web server for DSS administration through a Web browser.

   - Create a DSS suffix for storing WebSphere Everyplace Connection Manager details.
   - Refer to Chapter 6., “IBM Directory Server configuration” on page 65 for further information.
5. Install WebSphere Everyplace Connection Manager packages.
6. Install the Gatekeeper.
7. If required, install and configure a proxy server. This will be required for all WAP services, and may also be used by non-WAP applications between WebSphere Everyplace Connection Manager and back-end applications.
8. Configure Access Manager
   - Start Gatekeeper.
   - Configure settings for DSS access.
   - Refer to Chapter 7., “Access Manager configuration” on page 75 for further details.
9. Using the Gatekeeper, create Connection Manager resources as defined by your solution requirements:
   - Configure persistent storage.
   - Configure necessary resources.

2.6 WebSphere Everyplace Suite considerations

When installing WebSphere Everyplace Connection Manager V5 in a WebSphere Everyplace Suite environment, there are some additional considerations that must be taken into account. The following section identifies these areas and discusses the techniques used to integrate WebSphere Everyplace Connection Manager and WebSphere Everyplace Suite V2.1.5.

2.6.1 Installation

At the time of this redbook, the current version of WebSphere Everyplace Suite is V2.1.5. The version of WebSphere Everyplace Connection Manager supplied with this product is version 4.2. The items discussed in this section, while based on V4.2, may also be applied to WebSphere Everyplace Connection Manager V5.

Typically, all components of a WebSphere Everyplace Suite environment are installed through the WebSphere Everyplace Suite installation manager found on the first CD of the product. This is not the case with WebSphere Everyplace Connection Manager. WebSphere Everyplace Connection Manager must be installed by its own installer, and not through the WebSphere Everyplace Suite installer.

The steps required to install WebSphere Everyplace Connection Manager in this environment are as follows:

1. As is the case with a standalone WebSphere Everyplace Connection Manager installation, the directory server will be installed first. Once the LDAP has been installed and configured for use by the WebSphere Everyplace Suite components, additional entries will need to be added to allow integration with WebSphere Everyplace Connection Manager. This may be done using the LDIF definition provided in “Sample LDIF” on page 584. Run the command **`ldapmodify`** to add the appropriate entries.
2. From the command prompt on the WebSphere Everyplace Connection Manager server, run the following command per your operating system:
   - AIX: **`mkdir -p /usr/lpp/wireless`**
   - Solaris: **`mkdir -p /opt/wireless`**
3. Change to the directory created in step 2 and create a file called wgmgrd.conf through a text editor. Using the sample supplied in “WebSphere Everyplace Suite samples” on page 583, add the information to this new file. Be sure to check all values, ensuring they match your environment. Save this file when complete.
4. Install and configure the following items supplied with WebSphere Everyplace Suite onto the WebSphere Everyplace Connection Manager server:
   - Database server for use by WebSphere Everyplace Connection Manager V5, such as DB2 or Oracle
   - GSKit for use with certificates and SSL
   - IBM Directory clients:
     - AIX: Install ldap.client and ldap.max_crypto_client
     - Solaris: Install IBMldapc and IBMldapdj
5. Now that the WebSphere Everyplace Suite environment and WebSphere Everyplace Connection Manager host has been prepared, WebSphere Everyplace Connection Manager may be installed per the standard installation instructions found in this redbook.

2.6.2 User management

In a WebSphere Everyplace Suite environment, user enrollment and management is typically done through WebSphere Everyplace Subscription Manager (WESM). The Gatekeeper uses two fields in determining if to display a user in a search result:

- ibm-WGClient: If `true`, this indicates the user is a WebSphere Everyplace Connection Manager user.
- ibm_WAPclient: Indicates a WAP user.

The Gatekeeper will display the users where the above value is set to `true`. When a user is enrolled through WESM, both of these values will be set to `true` by default. This may be controlled by the properties file used by the WESM LDAP provisioning gateway, LDAPgateway properties. This may be modified as required in order to correctly configure as they are created.

WebSphere Everyplace Suite users will not be shown in the Gatekeeper as WebSphere Everyplace Connection Manager users until either one of the following occurs:

- A WebSphere Everyplace Suite user connects to the Connection Manager.
- A change is made in the Gatekeeper to any of the properties for a WebSphere Everyplace Suite user.

Either one of the above scenarios will allow WebSphere Everyplace Connection Manager to add the required user fields to a WebSphere Everyplace Suite user to indicate it is also a valid WebSphere Everyplace Connection Manager user.

3

AIX installation

This chapter provides detailed instructions for installing and configuring WebSphere Everyplace Connection Manager for AIX. The topics covered in this chapter include:

- Preparation of the AIX environment
- Verification of prerequisite software
- Installation of WebSphere Everyplace Connection Manager v5
- Installation verification

The process described within this chapter details the steps required for a first-time install onto the system.

Note: Also see Appendix A, “AIX: Step-by-step installation” on page 529.

3.1 Configuring the operating system

The details in this chapter assume the AIX operating system has already been installed. While configuration issues will be discussed, the installation process for the operating system will not be covered.

For further details regarding installation of AIX, consult your system administrator or the relevant IBM documentation available from:

http://www.ibm.com

3.1.1 Verifying the operating system level

WebSphere Everyplace Connection Manager v5 requires the operating system to be AIX 5.1 as a minimum. The operating system version may be found with the **oslevel** command as shown in Example 3-1. This indicates the operating environment release is V5.2.0.0. This is the output from the host used for installation and testing during the production of this chapter.

Example 3-1 Operating system version

```
# oslevel
5.2.0.0
#
```

3.1.2 Verifying the hardware requirements

The minimum hardware requirement on an AIX platform is a 375 MHz processor and 256 MB RAM available. To ensure the system is suitable, some commands may be used to check the details. First, use the command **lsdev -C |grep proc** to see how many processors the equipment has (as in Example 3-2). This command displays information about all devices in the customized devices object class with the name, status, location, and description.

Example 3-2 Obtaining the number of processors

```
# lsdev -C |grep proc
proc0      Available 00-00       Processor
#
```

After identifying the processors, the capacity of each processor can be displayed. This output was obtained from the host used for installation and testing during the production of this chapter. This host has only one processor named proc0. Using the command **lsattr -El proc0**, the capacity of the processor will be displayed as in Example 3-3. This command will display the processor state, type, and frequency. The one in the example has 375Mhz.

Example 3-3 Obtaining the processor frequency

```
# lsattr -El proc0
state     enable       Processor state False
type      PowerPC_604e Processor type  False
frequency 375000000    Processor Speed False
#
```

The second step is to recognize the RAM memory available. As done with its processors, you will first need to find how many memory devices the system has. The command **`lsdev -C |grep mem`** will list this information for you as shown in Example 3-4. This command displays information about all the devices in the customized devices object class with the name, status, location, and description.

Example 3-4 Obtaining the number of memory devices

```
# lsdev -C |grep mem
mem0       Available              Memory
#
```

Once you have identified all memory devices, you can also display their capacity. The host system used for this example has only one memory device named `mem0`. Using the command **`lsattr -El mem0`** now, the capacity of the memory will be displayed as in Example 3-5. This command will display the memory size and the amount that can be used.

Example 3-5 Obtaining the memory size

```
# lsattr -El mem0
size     512 Total amount of physical memory in Mbytes  False
goodsize 512 Amount of usable physical memory in Mbytes False
#
```

If these requirements are attended, the installation process can proceed.

3.2 Installation tasks

This section describes the process for installing WebSphere Everyplace Connection Manager v5 on an AIX system.

3.2.1 Pre-installation tasks

Prior to installing WebSphere Everyplace Connection Manager v5, the following items must be verified and completed:

1. One of the following database environments is installed for use by the Connection Manager for the persistent storage of session data:

   – DB2 Universal Database™ Enterprise edition v7.2

   – Oracle Version 8.1.5, 8.1.6, or 8.1.7, which also requires Merant DataDirect Connect ODBC V4.2.

   Important: When Using DB2, the /etc/environment file must be repaired. Usually, the DB2 installation sets the Java default to a back level. To repair it, edit this file and move the two java130 statements to the front of the PATH. Then log out and log back in.

2. One of the following directory servers is installed for use by the Connection Manager for persistent storage on configuration information:

   – IBM SecureWay® Directory 3.2.1
   – IBM Directory Server 4.1
   – Netscape Directory Version 4.1x
   – iPlanet Directory Server 5.x

3. TCP/IP connectivity is active. Use commands such as **`ping`** and **`ifconfig -a`** to ensure all relevant hosts within the environment are reachable and configured correctly.

4. If WebSphere Everyplace Connection Manager is not being installed on the same hosts as the directory server, the IBM Key Management tool and the IBM Directory Client may need to be installed. These products are installed by default with IBM Directory Server. Use the following methods to check for each product:

   – Check for IBM Key Management tool using the following command:

   ```
   # lslpp -l |grep gskkm
   ```

   If IBM Key Management is installed, the output will look similar to Example 3-6.

Example 3-6 Check for gskkm

```
# lslpp -l|grep gskkm
  gskkm.rte                 5.0.5.79  COMMITTED  AIX Certificate and SSL Base
```

If no output is returned, refer to “Installing Directory service client” on page 41.

Installing IBM Key Management

On AIX, the IBM Key Management tool (gskit) can be automatically installed with Connection Manager. If you prefer the automatic installation, is not necessary follow this topic. Anyway, it can be installed using the following steps prior to installing Connection Manager:

1. Locate the gskkm.rte package provided with the Connection Manager software:

   ```
   # cd <cdrom_path>/usr/sys/inst.images
   ```

2. Use installp_cmd to install:

   ```
   # /usr/lib/instl/sm_inst installp_cmd -a -Q -d '.' -f 'gskkm'
   ```

Installing Directory service client

On AIX, the IBM Directory service can be automatically installed with Connection Manager. If you prefer the automatic installation, is not necessary follow this topic. Anyway, it can be installed using the following steps prior to installing Connection Manager:

1. Locate the ldap.client package provided with the Connection Manager software:

   ```
   # cd <cdrom_path>/usr/sys/inst.images
   ```

2. Use installlp_cmd to install:

   ```
   # /usr/lib/instl/sm_inst installp_cmd -a -l -d
   '.<cdrom_path>/use/usr/sys/inst.images' -f 'ldap.client'
   ```

3.2.2 Connection Manager and Gatekeeper installation tasks

The following section describes the steps involved in installing the Connection Manager component. On AIX, the smitty tool can do the installation of all relevant packages necessary. For example:

1. Make sure that you have installed the Directory service client (on the same system).
2. Locate the WebSphere Everyplace Connection Manager v5:

   ```
   # cd <cdrom_path>/usr/sys/inst.images
   ```

3. Invoke smitty

   ```
   # smitty
   ```

4. Select this path through the menus:

   a. Software installation and maintenance
   b. Install and update software.

c. Install and update from latest available software.

5. Fill the field INPUT device/directory for software with “ **.**”

   [.]

6. For the SOFTWARE to Install field, press F4 to list the components.
7. Select the components you want to install using F7:
   - All wg components install the Connection Manager code.
   - wgcfg is the Gatekeeper code. Select it just if you want to administer the Gatekeeper on the same machine.
   - ldap.client is the directory service client.
   - gskkm is the IBM Key Management.
8. Press Enter three times.

4

Solaris installation

This chapter provides detailed instruction for installing and configuring WebSphere Everyplace Connection Manager for Solaris. The topics covered in this chapter include:

- Preparation of the Solaris environment
- Verification of prerequisite software
- Installation of WebSphere Everyplace Connection Manager v5
- Installation verification

4.1 Hardware requirements

The following shows the minimum hardware requirements for running Connection Manager V5 on a Solaris platform:

- Ultra 10
- 1 GB RAM
- Minimum 9 GB HDD

4.2 Configuring the operating system

The details discussed in this chapter assume the Solaris operating system has already been installed. While configuration issues will be discussed, the installation process for the operating system will not be covered.

For further details regarding installation of Solaris, consult your system administrator or the relevant Sun documentation available from:
http://www.sun.com

4.2.1 Verifying the operating system level

WebSphere Everyplace Connection Manager v5 is supported on the following Sun Solaris operating systems:

- Solaris V7
- Solaris V8
- Trusted Solaris V8

The operating system version may be found with the **uname -a** command as shown in Example 4-1. This indicates the operating environment is SunOS Release 5.8 (Solaris 8), the host name is sun4, with the current kernel version, running on Sparc Ultra-60 hardware. This is the output from the host used for installation and testing during the production of this chapter, and meets all minimum requirements for operating the Connection Manager and related products.

Example 4-1 Operating system version

```
# uname -a
SunOS sun4 5.8 Generic_108528-22 sun4u sparc SUNW,Ultra-60
```

The **showrev** command used in Example 4-2 shows similar details relating to the operating environment.

Example 4-2 Output from showrev

```
# showrev
Hostname: sun4
Hostid: 808a622c
Release: 5.8
Kernel architecture: sun4u
Application architecture: sparc
Hardware provider: Sun_Microsystems
Domain:
Kernel version: SunOS 5.8 Generic 108528-22 May 2003
```

If is recommended that the operating system is at the highest available patch level and kernel revision. To ensure this, download and install the latest recommended patch cluster from:
http://sunsolve.sun.com.

4.3 Installation tasks

This section describes the process for installing WebSphere Everyplace Connection Manager v5 on a Solaris system.

4.3.1 Pre installation tasks

Prior to installing WebSphere Everyplace Connection Manger v5, the following items must be verified and completed:

1. One of the following database environments is installed for use by the Connection Manager for persistent storage of session data:
   - DB2 Universal Database Enterprise Edition V7.2
   - Oracle Version 8.1.5, 8.1.6, or 8.1.7 with MercantDataDirect Connect ODBC V4.2

   > **Note:** When using DB2, ensure the kernel parameters in /etc/system have been updated as per the details in /opt/IBMdb2/V7.1/cfg/kernel.param.*MB, where * is the amount of memory in the system. Any change to /etc/system will require a reboot before taking effect.

2. One of the following directory servers is installed for use by the Connection Manager for persistent storage on configuration information:
   - IBM SecureWay Directory 3.2.1

   - IBM Directory 4.1
   - Netscape Directory Version 4.1x.
   - iPlanet Directory Server 5.x

3. TCP/IP connectivity is active. Use commands such as **`ping`** and **`ifconfig -a`** to ensure all relevant hosts within the environment are reachable and configured correctly.

4. If WebSphere Everyplace Connection Manager is not being installed on the same hosts as the directory server, the IBM Key Management tool and the IBM Directory client may need to be installed prior to installing Connection Manager. These products are installed by default with IBM Directory Server. Use the following methods to check for each product:

   - Check for the IBM Key Management tool (GSKit) using the following command:

     ```
     # pkginfo -l|grep gsk5bas
     ```

     If IBM Key Management is installed, the output will look similar to Example 4-3.

Example 4-3 Check for gsk5bas

```
# pkginfo -l |grep gsk5bas
   PKGINST:  gsk5bas
      NAME:  Certificate and SSL Base Runtime (gsk5bas)
#
```

     If no output is returned, refer to “Installing IBM Key Management” on page 46 for installation details.

   - Check for the IBM Directory client using the following command:

     ```
     # pkgadd -l|grep IBMldapc
     ```

     If no output is returned, refer to “Installing IBM directory client” on page 47 for installation details.

Installing IBM Key Management

On Solaris, the IBM Key Management tool is not automatically installed with Connection Manager. Use the following steps to install this prior to installing Connection Manager:

1. Locate the gsk5bas package provided with the Connection Manager software:

   ```
   # cd <cdrom_path>/solaris
   ```

2. Use **`pkgadd`** to install:

   ```
   # pkgadd -d gsk5bas
   ```

3. Select package **1** to install gsk5bas as shown in Example 4-4.

Example 4-4 IBM Key Management install

```
The following packages are available:
  1  gsk5bas     Certificate and SSL Base Runtime (gsk5bas)
                 (sparc) 5.0.5.79

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: 1
```

4. Press Y to allow the installation to continue with super user permission.

Installing IBM directory client

The IBM Directory client is not automatically installed. Perform the following steps to install prior to installing Connection Manager:

1. Locate the IBMldapc package:

   ```
   # cd <cdrom_path>/solaris
   ```

2. Use **pkgadd** to install:

   ```
   # pkgadd -d IBMldapc
   ```

3. Select package**1** to install the IBMldapc package.
4. When prompted, confirm the installation directory by pressing Enter to accept the default, /opt, or enter an alternate path.
5. Press Y to allow the installation to continue with super user permission.
6. Change to the directory where the directory client was installed to confirm success.

4.3.2 Connection Manager installation tasks

The following section describes the steps involved in installing the Connection Manager component. On Solaris, a script is provided for this purpose to assist with adding the relevant packages necessary for the install. Table 4-1 shows the packages included for Solaris.

Table 4-1 Solaris packages

Package Name	Description
IBMwgrte	Connection Manager run time and Access Manager support
IBMwgdrad	Dataradio
IBMwgdtac	DataTAC

Package Name	Description
IBMwgdial	Dial up
IBMwgipla	IP-LAN
IBMwgiwla	IP-LAN with WTLS
IBMwgmobi	Mobitex
IBMwgrdis	Motient
IBMwgppg	Messaging Services
IBMwgrnc3	RNC-3000
IBMwgsms	Short Message Service
IBMwgsmtp	Simple Mail Transport Protocal
IBMwgsnpp	Simple Network Paging Protocol
IBMwgwap	WAP proxy
IBMwgwctp	Wireless Communication Transfer Protocol
IBMwgwlp	Mobile Access Service
IBMwgnls	National language version support

Depending on the requirements of the system, any combination of packages may be installed, however, IBMwgrte is required by all other packages.

Note: The WAP proxy is only supplied with Connection Manager V5 Starter Edition and Connection Manager V5 Full.

For the installation of the Connection Manager, complete the following steps:

1. Locate the Connection Manager installation image directory:

   ```
   # cd <path>/solaris
   ```

2. Type **./install_wg** and press Enter.
3. The list of packages is displayed. Take note of the package numbers required for the install, and enter the details when requested, similar to Example 4-5.

Example 4-5 Package selection

```
The following packages are available:
  1  IBMwg2rte     Connection Manager Run-time Environment
                   (sparc) 5.0.0.0
```

```
  2  IBMwgdial     Connection Manager Dial Support
                   (sparc) 5.0.0.0
  3  IBMwgdrad     Connection Manager Dataradio Support
                   (sparc) 5.0.0.0
  4  IBMwgdtac     Connection Manager DataTAC Support
                   (sparc) 5.0.0.0
  5  IBMwgipla     Connection Manager IP-LAN Support
                   (sparc) 5.0.0.0
  6  IBMwgiwla     Connection Manager IP-WTLS-LAN Support
                   (sparc) 5.0.0.0
  7  IBMwgmobi     Connection Manager Mobitex Support
                   (sparc) 5.0.0.0
  8  IBMwgmoda     Connection Manager Modacom Support
                   (sparc) 5.0.0.0
  9  IBMwgrdis     Connection Manager Ardis Support
                   (sparc) 5.0.0.0
 10  IBMwgrnc3     Connection Manager RNC-3000 Support
                   (sparc) 5.0.0.0

... 4 more menu choices to follow;
<RETURN> for more choices, <CTRL-D> to stop display:

 11  IBMwgsms      Connection Manager Short Message Service
                   (sparc) 5.0.0.0
 12  IBMwgsmtp     Connection Manager SMTP Email Support
                   (sparc) 5.0.0.0
 13  IBMwgsnpp     Connection Manager SNPP Paging Support
                   (sparc) 5.0.0.0
 14  IBMwgwctp     Connection Manager WCTP Paging Support
                   (sparc) 5.0.0.0

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: 1,5,7
```

Note: Not all available packages are shown. Prompts will appear towards the end of the install to confirm installation of IBMwgwap, IBMwgwlp, and IBMwgppg packages.

4. When prompted, enter this installation directory, or press Enter to accept the default value of /opt.
5. Press Y to allow the process to run with super user permission.
6. When prompted, select whether or not the remaining three packages are to be installed.
7. The process is complete. You may repeat these steps at any time in order to add additional packages to the Connection Manager install.

4.3.3 Gatekeeper installation tasks

The following steps describe the tasks required to install the Gatekeeper component used for managing Connection Manager resources. If a previous version of the Gatekeeper is already installed on the system, remove it using the **`pkgrm IBMwgcfg`** command, and delete .wgcfg from each user's home directory.

1. Locate the Connection Manager installation image directory:

   ```
   # cd <path>/solaris
   ```

2. Type **`pkgadd -d IBMwgcfg.solaris.pkg`** and press Enter.
3. Select the Gatekeeper package.
4. When prompted, enter this installation directory, or press Enter to accept the default value of /opt.
5. Allow the install to run with super-user permission.
6. When complete, the Gatekeeper can be started by entering **`wgcfg`** at a command prompt.

Refer to Chapter 7., "Access Manager configuration" on page 75 for details on using the Gatekeeper to complete the Connection Manager configuration.

5

Linux installation

Confirming the approach towards Linux, IBM WebSphere Everyplace Connection Manager can also run on this platform. This chapter describes a sample setup, and the installation and configuration required when installing Connection Manager on a Linux system running the Red Hat operating system.

The following sections are included in this chapter:

- Planning
- System requirements
- Database and LDAP configuration
- WebSphere Everyplace Connection Manager installation
- Gatekeeper installation
- Basic configuration

5.1 Planning

WebSphere Everyplace Connection Manager relies on other prerequisites products to run. Other than the Linux operating system, the user must understand the need of other products such as:

- Relational database
- LDAP Directory Services

Before starting the installation, it should be considered where the database and LDAP Directory will be located. These products will be used to store session data, and optionally accounting and billing data of WebSphere Everyplace Connection Manager.

There are two basic approaches:

- Install them on the same box as WebSphere Everyplace Connection Manager.
- Install them on a different box from where WebSphere Everyplace Connection Manager will be installed. This approach requires the installation of a LDAP client and a DB2 client.

The first approach was chosen for the sample scenario as shown in Figure 5-1.

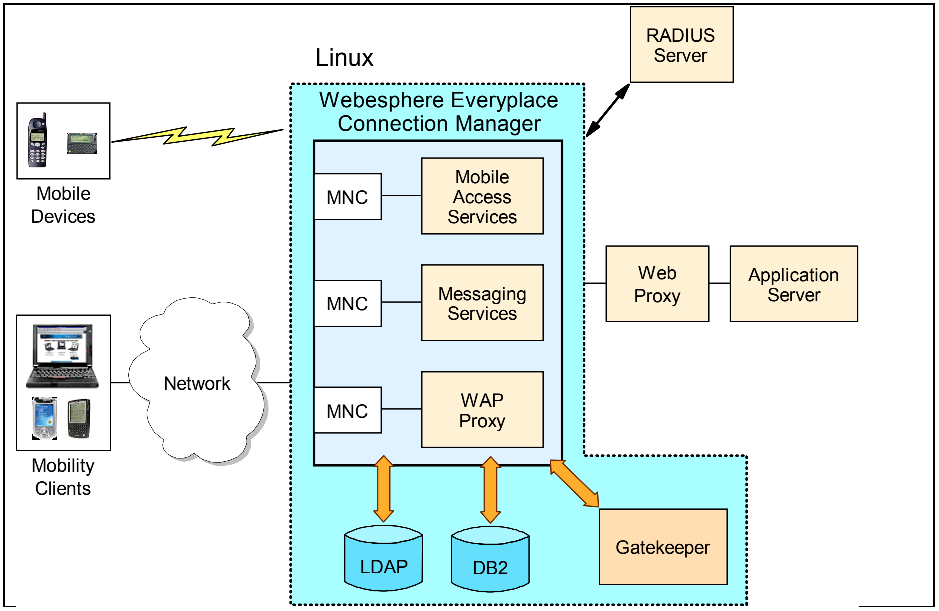


Figure 5-1 Everyplace Connection Manager sample installation on Linux platform

Table 5-1 shows the software components and their versions, which will be used for this scenario.

Table 5-1 Sample software composition for Connection Manager installation

Component	Product	Version (fix level)
Operating system	Linux Red Hat	8.0
LDAP Server	IBM Directory Sever	5.1
DB Server	IBM DB2 UDB for Linux	8.1(2)
WebSphere Everyplace Connection Manager	Connection Manager, Gatekeeper	5.0.1.1

With this installation, you will have following components:

- Messaging Services

  Enables a Web application server to send messages to messaging clients, such as a pager, or a mobile phone using a variety of wireless networks. For more information, refer to Chapter 18, “Messaging services” on page 371.

- Mobile Access Services

  Creates an optimized and secure IP tunnel to communicate with any device using the Mobility Client software. The Mobility Client can use a variety of wireless and wireline devices connecting them to the company’s private intranet or to the Internet. For more information, refer to Chapter 9, “Mobile Access Services” on page 137.

- WAP proxy

  Performs a protocol conversion between HyperText Transport Protocol (HTTP) and Wireless Application Protocol (WAP) protocols to link WAP clients with Web-based browse services or TCP application services. For more information including configuring WAP proxy, refer to Chapter 19, “WAP gateway” on page 415.

- Gatekeeper

  An easy-to-use administrative interface that enables you to define and manage wireless resources. For more information about Gatekeeper, refer to Chapter 8.1, “Administration using Gatekeeper” on page 86.

5.2 System requirements

It is assumed that the Linux operating system was successfully installed, and the basic network functions are operational. Although it is not the purpose of this redbook to describe the installation of a Linux distribution, see 5.2.3, “Sample Linux installation” on page 55 regarding the steps followed to install Linux for this scenario. For more details regarding Linux installation, consult your system administrator or refer to more specific documentation, but keep in mind that Connection Manager hardware and software requirements must be fulfilled.

5.2.1 Hardware requirements

The following hardware is required to use the Connection Manager on Linux distribution:

- x86 platform server with a minimum of a 400MHz processor
- 256 MB RAM (if you install Relational Database Server which will be used for managing Connection Manager resources on same box with Connection Manager, you need at least 512 MB RAM)
- 100 MB disk space available (200 MB recommended)

5.2.2 Software requirements

WebSphere Everyplace Connection Manager was tested and certified for several Linux distributions and versions. Here is the list released by the time this documentation was generated.

Table 5-2 Supported Linux distributions

Distribution	Kernel level	glibc level
Red Hat 7.3	2.4.18	glib 2.2.5
Red Hat 8.0	2.4.18	Ships with glib 2.2.93
SuSE 7.3	2.4.10-4gb or later versions available from SuSE FTP site	Patched glibc 2.2.4 available from yast2 online update
SuSE 8.0	2.4.18	glib 2.2.5
Red Hat Enterprise Linux 3.0 ES		

Note: To display the current kernel level, enter the command from a command line:

```
uname -a
```

To display the current glibc level enter from a command line:

```
rpm -q glibc
```

5.2.3 Sample Linux installation

In this section we include a sample step-by-step Red Hat Linux installation. For example:

1. Power on the machine.
2. Make sure the BIOS is set to boot from CD. Save the configuration if it is necessary.
3. Power off the machine.
4. Insert the Red Hat Linux CD 1.
5. Power on the machine.
6. Wait for a text menu.
7. To install or upgrade Red Hat Linux in graphical mode, press the Enter key.
8. On the Welcome Red Hat Linux screen, click **Next**.
9. On the Language Selection screen, select **English** (English) and click **Next**.
10. On the Keyboard Configuration screen, select U.S. English and click **Next**.
11. On the Mouse Configuration screen, select **2 button mouse** (PS/2®) and click **Next**.
12. On the Installation Type screen, select **Custom** and click **Next**.
13. On the Disk Partitioning Setup screen, select the **Automatically Partition** option and click **Next**.
14. On the Automatic Partitioning screen, select the **Remove all partitions on the system** option and click **Next**.
15. On the Disk Setup screen, click **Next**.
16. On the Boot Loader Configuration screen, click **Next**.
17. On the Network Configuration screen, click **Edit** and uncheck the Configuring using DHCP option.
18. Fill out the IP address and the netmask. Click **OK.**
19. Fill out the hostname, gateway, and DNS. Click **Next**.
20. On the Firewall Configuration screen, check the **No Firewall** option, and click **Next.**
21. On the Additional Language Support screen, click **Next**.
22. On the Time Zone Selection screen, select a proper time zone, and click **Next.**
23. On the Account Configuration screen, type in the root password, confirm it, then click **Next**.

24. On the Authentication Configuration screen, click **Next**.
25. On the Package Group Selection screen, select packages according next topic, and click **Next**.
26. On the About to Install screen, click **Next**.
27. Change the CDs as asked for, and click **OK**.
28. Create a boot diskette
29. On the *G*raphical Interface Configuration screen, select a proper configuration if it is not detected automatically, then click **Next**.
30. On the Monitor Configuration screen, select a proper configuration if it is not detected automatically, then click **Next**.
31. On the Customize Graphics Configuration screen, select a proper configuration if it is not detected automatically, then click **Next**.
32. Click **Finish**.

Linux installation guidelines

The following general guidelines should be considered when installing the Linux operating system:

- Partitions

  There are many ways to do the disk partition. But there is a rule to obey, do not leave less than 3 GB, preferably 6 GB for the / partition.

- Boot Loader

  Make sure GRUB is selected. It will be important for upgrade purposes.

- Network configuration

  It is a good approach to use a static IP address and a normal host name this will make it easier using and setting up the gateway. DHCP is also allowed and can be used if necessary.

- Package group selection

  This is the selection for the sample scenario used in this chapter:

  – Desktops

    • X Window system
    • GNOME desktop environment
    • KDE desktop environment

  – Applications

    • Editors
    • Graphical Internet
    • Text-base Internet

- Servers
  - Server configuration tool
  - Windows file server
  - FTP server
  - Network servers
- Development
  - Development tools
  - Kernel development
  - X Software Development
  - GNOME software development
  - KDE software development
- System
  - Administration tools
  - System tools

5.3 Database and LDAP configuration

This topic covers all the configurations that need to be set up before installing the WebSphere Everyplace Connection Manager.

5.3.1 Preinstallation task

This chapter does not approach the installation of DB2 Database or Directory Server, it is assumed that they were installed without errors or warnings, and that they are ready to be configured to work with WebSphere Everyplace Connection Manager. For detailed installation steps, refer to the DB2 installation guide and Directory Server installation guides from IBM Web site.

IBM DB2 configuration

The Relational Database (RDB) environment is installed to be used by the Connection Manager for the persistent storage of session data, and optionally for billing and accounting purposes.

Before installing WebSphere Everyplace Connection Manager, it is necessary to check the group file to make sure the DB2 user groups have the proper privileges.

Check the group file in the /etc directory. For all users created while you install DB2, check if those user's group have a root user as well.

Your group file must be similar to Example 5-1.

Example 5-1 User group file

```
db2fadm1:x:102:
db2iadm1:x:101:db2as,root
db2asgrp:x:103:ldapdb2,wgdb
db2:x:501:root,db2as
ldap:x:200:root,root
```

If not, add a root user to a group such as in Example 5-1.

> **Important:** If you install IBM DB2 Server on same Linux box with Connection Manager, and use the Linux 2.4 kernel environment, you must change the default value for msgmni from 16 to a default value of 1024 or higher.
>
> Configure the msgmni parameter by issuing the **sysctl** command as root:
>
> ```
> #sysctl -w kernel.msgmni=1024
> ```
>
> To set the msgmni kernel parameter at boot time, append the following lines to /etc/sysctl.conf:
>
> ```
> # Sets maximum number of message queues to 128
> # Set this to 1024 or higher on production systems
> kernel.msgmni = 1024
> ```
>
> If you do not change this parameter, you may encounter serious problems starting your DB2 instance. For more information, refer to:
> http://www.ibiblio.org

IBM Directory Server (LDAP) configuration

The Light Weight Directory Server environment is installed to store configuration data of WebSphere Everyplace Connection Manager. To install WebSphere Everyplace Connection Manager, perform following tasks:

1. With the **ldapxcfg** command, perform the initial configuration step. For detailed steps, refer to Chapter 6.1.1, “Configuring with ldapxcfg” on page 66.

   Except for the configuration option for a Web server, all configuration steps are the same with the IBM Directory Configuration chapter. Note that on the Linux platform, IBM Directory Server does not provide a Web administration GUI. You must use the command line commands to manage the LDAP server.

2. Now, we will add a suffix for WebSphere Everyplace Connection Manager. On the Linux platform, you have to use command line commands to create a suffix. Add the following line **ibm-slapdsuffix:o=ibm,c=us** to the

ibmslapd.conf file stanza such as in Example 5-2. The file resides in the /etc directory.

Example 5-2 Adding a suffix on LDAP server

```
DN:cn=Directory cn=RDBM Backends cn=IBM SecureWay cn=Schemas cn=Configuration
ibm-slapdDbName:<databasename >
ibm-slapdUserID:<username >
ibm-slapdDbUserPW:<password >
ibm-slapdDbInstance:<username >
ibm-slapdSuffix:o=ibm,c=us
```

3. If LDAP started, stop and restart LDAP server. You can stop and start LDAP server with the following command:
   - Stop: find out the process of ldap by using the command:

     `ps -ef |grep slapd`

     and kill that process.
   - Start: type **`slapd`**.

Note: With IBM Directory Server 5.1, instead of editing the ibmslapd.conf file, you can use the command on command line. Type the following command to add your suffix (for example, `o=ibm,c=us`).

```
ldapcfg -s o=ibm,c=us
```

With IBM Directory Server 5.1 on a Linux platform, you need to start LDAP server with following command:

```
ibmslapd -h 0
```

This starts LDAP server with debug mode, which produces a debug log file with level 0. Otherwise, the LDAP server will crash on the Linux platform.

Other environment check

The following guidelines should be considered:

1. TCP/IP connectivity is active. Use a command such as **`ping`** and **`ifconfig -a`** to ensure all relevant hosts within the environment are reachable and configured correctly. Make sure that the hostname for the machine is not aliased in the /etc/hosts file to the loopback address.
2. Gatekeeper requires the IBM JRE Version 1.3.1 service, release 3 (and all of the things that the IBM JRE 1.3.1 requires). On the installation CD in the /gatekeeper/linux directory is the file IBM Java2-JRE-1.3.1-3.0.i386.rpm. This

file is the IBM JRE1.3.1. Or, you can download it from the IBM Support Web site.

3. To provide mobile network interface support, the kernel source and kernel headers are required. For example, on a Red Hat 7.3 distribution, the kernel-2.4.18-14 and kernel-source-2.4.18-14 packages must be installed. To display the current kernel level, from the command line enter:

   ```
   uname -a
   ```

   You can install kernel packages from your Linux installation CD, or download it from the Linux Web site.

Installing IBM Key Management

On the Linux system, IBM Key Management is not automatically installed. Perform the following steps to install prior to installing Connection Manager:

1. Locate the gskit package provided with WebSphere Connection Manager software:

   ```
   # cd <cdrom_path>/linux
   ```

   There are two rpm packages for gskit.

   ```
   gsk5bas-5.0.5-79.i386.rpm
   gsk5ikm-5.0.5-79.i386.rpm
   ```

2. Use the **rpm** command to install the gsk packages.

   Installation of gsk packages looks like Example 5-3.

Example 5-3 installation of IBM Key Management

```
# rpm -ivh gsk5bas-5.0-5.79.i386.rpm
Preparing...                ########################################### [100%]
   1:gsk5bas                ########################################### [100%]
# rpm -ivh gsk5ikm-5.0-5.79.i386.rpm
Preparing...                ########################################### [100%]
   1:gsk5ikm                ########################################### [100%]
```

Installing IBM Directory Client

If you install WebSphere Everyplace Connection Manager on a different machine from Directory Server, the IBM Directory client has to be installed. Perform the following steps to install prior to installing Connection Manager:

1. Locate the IBM Directory Client package provided with WebSphere Everyplace Connection Manager software:

   ```
   # cd <cdrom_path>/linux
   ```

   There are two packages regarding IBM Directory Client module:

   ```
      ldap-client-4.1-1.i386.rpm
   ```

```
ldap-msg_XXXX-4.1-1.i386.rpm
```

where xxx is the selected language.

2. Use the **rpm** command to install IBM Directory Client.

   Installation of IBM Directory Client looks like Example 5-4.

Example 5-4 installation of IBM Directory Client

```
# rpm -ivh ldap-client-4.1-1.i386.rpm
Preparing...                ########################################### [100%]
   1:ldap-client            ########################################### [100%]
# rpm -ivh ldap-msg_en_US-4.1-1.i386.rpm
Preparing...                ########################################### [100%]
   1:ldap-msg_en_US         ########################################### [100%]
```

Note: IBM Directory runtime client may conflict with the existing installation packages. Make sure that the nss_ldap package is not installed before installing Connection Manager.

5.4 Connection Manager installation

This section describes a sample WebSphere Everyplace Connection Manager installation on a Linux platform.

To install WebSphere Everyplace Connection Manager, perform following steps:

1. Insert the installation CD into the CD drive and change the directory to /linux.
2. Enter the following command:

   ./install_wg

Example 5-5 Installation of Connection Manager on Linux

```
# ./install_wg
The Connection Manager requires either DB2 client version 7.1.0.40 (or newer)
or Oracle client version 8.1.5.0.0 (or newer)
Checking for DB2 client
DB2 client db2rte71 was found on this machine and will be used by the
Connection Manager
Backing up configuration files
Installing base Connection Manager Package...
########################################### [100%]
########################################### [100%]
creating /var/adm dir
Restoring previous configuration files.
```

3. On a Linux system, you are prompted whether to disable auto start at system boot. The default is that the Connection Manager will start at the system at boot time. Press N such as in Example 5-6.

Example 5-6 Select enabling gateway auto-start during installation

```
Disable gateway auto-start at system boot?[n]
n
```

4. You are prompted for the components or networks you want to install. For individual support packages, refer to the installation of Connection Manager on a Solaris platform. In this sample scenario, we selected all packages. See Example 5-7.

Example 5-7 Selecting installation packages

```
1.  IBMwg-ardis - Connection Manager Ardis support
2.  IBMwg-dataradio - Connection Manager Dataradio support
3.  IBMwg-datatac - Connection Manager DataTAC support
4.  IBMwg-dial - Connection Manager Dial support
5.  IBMwg-ip-lan - Connection Manager IP LAN support
6.  IBMwg-ip-wtls-lan - Connection Manager IP WTLS LAN support
7.  IBMwg-mobitex - Connection Manager Mobitex support
8.  IBMwg-modacom - Connection Manager Modacom support
9.  IBMwg-nls - Connection Manager National Language support
10.  IBMwg-ppg - Messaging Services
11.  IBMwg-rnc3000 - Connection Manager RNC3000 support
12.  IBMwg-sms - Connection Manager SMS support
13.  IBMwg-smtp - Connection Manager SMTP support
14.  IBMwg-snpp - Connection Manager SNPP support
15.  IBMwg-wap - Connection Manager WAP Proxy
16.  IBMwg-wctp - Connection Manager WCTP support
17.  IBMwg-wlp - Connection Manager Mobile Access

Enter the numbers of the additional packages you want to install
separated by spaces, or hit enter to install all packages.

########################################### [100%]
########################################### [  5%]
########################################### [ 11%]
########################################### [ 17%]
########################################### [ 23%]
########################################### [ 29%]
```

5. Now you are prompted to install Mobile Network Interface (MNI) support. This requires to compile your Linux kernel source. See Example 5-8.

Example 5-8 Specifying kernel source location to install MNI support

```
Now Mobile Network Interface (MNI) support will be installed
This part of the install requires Linux kernel headers.  If you
have not installed them, this part of the install can be run
at another time.  Just run "./install_mni" from the
"/opt/wireless/mnidev" directory.

Would you like to continue with the install[y]?
y
You are running kernel version 2.4.18-3 on Red Hat Linux release 7.3
(Valhalla).


Would you like to install MNI support for this kernel version[y]?
y

Please enter the path to the kernel source[/lib/modules/2.4.18-3/build].

rm -rf wg_mni.o
gcc -Wall -DMODULE -D__KERNEL__ -DLINUX -O2
-I/lib/modules/2.4.18-3/build/includ
e -c wg_mni.c -DWECM_GATEWAY=1 -o wg_mni.o
install -D -m 644 ./wg_mni.o /lib/modules/2.4.18-3/misc/wg_mni.o
/sbin/depmod -a 2.4.18-3 2> /dev/null > /dev/null
make: [install] Error 1 (ignored)
```

> **Note:** This process only needs to be completed one time. If you change or upgrade the kernel, you will need to perform this process again. The script to provide Mobile Network Interface (MNI) support is **`/opt/wclient/mnidev/install_mni`**.

5.5 Gatekeeper installation

The following steps describe how to install Gatekeeper used for managing Connection Manager resources. If you have previously installed the Gatekeeper, uninstall it. Note that the removal procedure from the Gatekeeper Version 4.2.1 and prior is different than version 4.2.2 and later. Uninstall the Gatekeeper using the removal procedure for the version you have installed. Also, delete the .wgcfg directory from each user's home directory.

The following is a sample installation procedure:

1. Log in as root on the system where you want to install the Gatekeeper.
2. Insert the installation CD into the CD drive and mount the CD.

3. From the /gatekeeper/linux directory, run the command:

   ```
   rpm -ivh IBMwgcfg-2.1-1.fixlevel.i386.rpm
   ```

   where `fixlevel` is the fix level designation of the product. This command installs the Gatekeeper to the default directory of /opt/wgcfg, and creates links in the /usr/bin directory for the following scripts:

   – **wfcfg**
   – **wgcfgikeyman**
   – **wgcfgnewjvm**
   – **wgcfgnewwin**

4. To start the Gatekeeper, run the script **wgcfg**.

5.6 Basic configuration

After installing WebSphere Everyplace Connection Manager and the Gatekeeper, it is time to perform an initial configuration of Connection Manager.

For example, perform the following steps:

1. Configure Access Manager through the Gatekeeper:

   For detailed steps, refer to Chapter 7.1, “The Access Manager” on page 76.

2. Add Connection Manager

   For detailed steps, refer to , “Adding Connection Manager” on page 94. After adding Connection Manager resource through Gatekeeper, you can start the gateway. There are two ways to start the gateway:

   – Through the Gatekeeper:

     Right-click your **Connection Manager resource** after logging into Gatekeeper. You can see the context menu for starting and stopping Connection Manager.

   – Command line:

     - To start type

       ```
       wg_start
       ```

     - To stop type:

       ```
       wg_stop
       ```

     - You can recognize the gateway process with the following command:

       ```
       ps -ef |grep wgated
       ```

3. Add other resources as required. Refer to 8.1.2, “Adding resources” on page 92.

6

IBM Directory Server configuration

This chapter describes the steps involved in configuring the IBM Directory Server for use with WebSphere Everyplace Connection Manager V5.

This chapter contains information on the following areas:

- Configuring the administrator’s ID
- Configuring the database
- Web server configuration
- Starting and stopping the directory server
- Adding a directory suffix

6.1 Configuring the Directory Server

The Directory Server may be configured using one of two different commands:

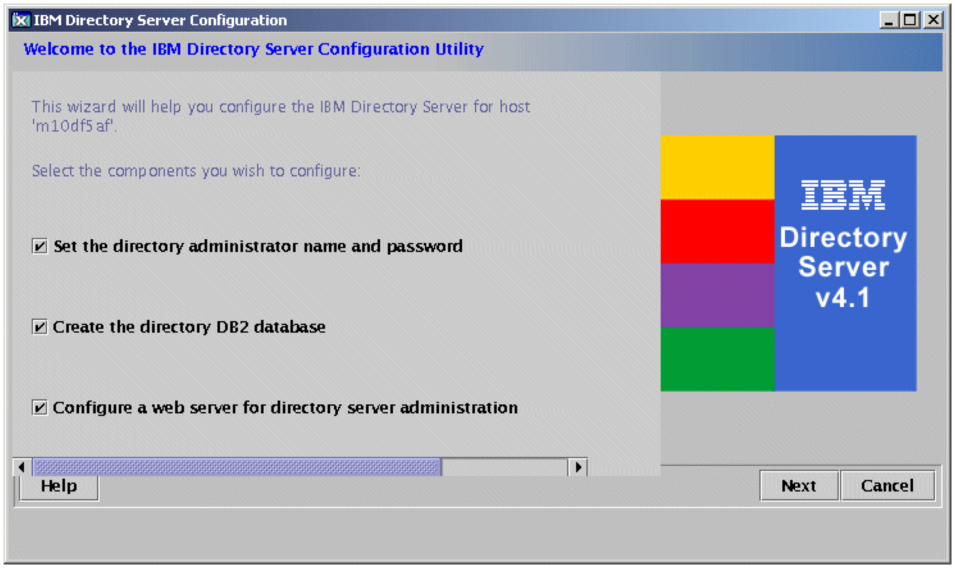
- **`ldapxcfg`**: This command provides a graphical interface for setting the administration ID and password, creating the Directory Server database, and configuring a Web server for the purpose of server administration.
- **`ldapcfg`**: This provides identical functionality to **`ldapxcfg`**, however, it runs from a command line interface. Enter **`ldapcfg`** at the command prompt to see the syntax required for this command.

6.1.1 Configuring with ldapxcfg

In order to use **`ldapxcfg`**, you must be using a system capable of displaying X-windows. Complete the following steps in order to configure the directory server:

1. To start the configuration utility, enter **`ldapxcfg`** at the command prompt.

> **Note:** Ensure your PATH variable includes /usr/bin. If not, run **`ldapxfg`** from with the /usr/bin directory. Typically, the PATH statement will be updated during the installation process.



IBM Directory Server Configuration

Welcome to the IBM Directory Server Configuration Utility

This wizard will help you configure the IBM Directory Server for host 'm10df5af'.

Select the components you wish to configure:

☑ Set the directory administrator name and password

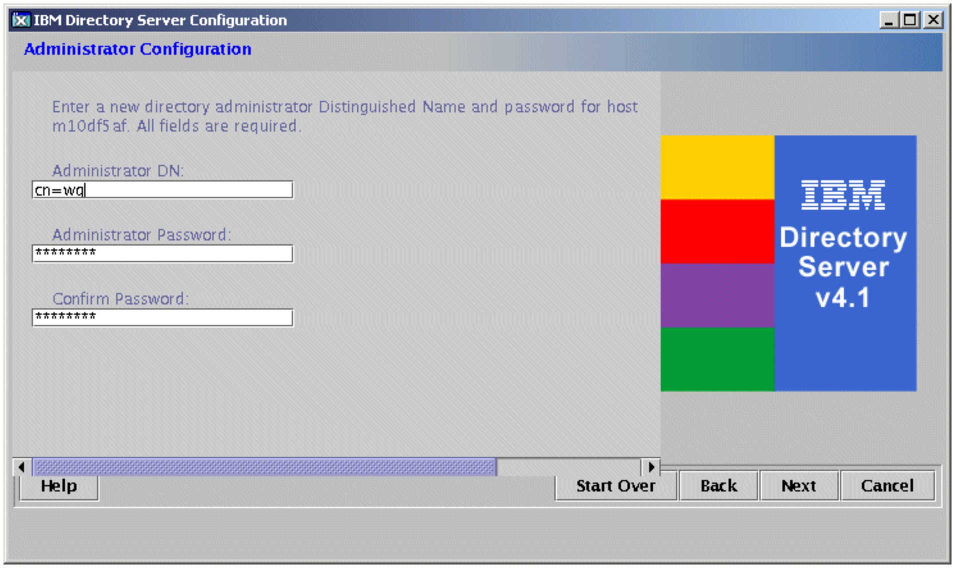
☑ Create the directory DB2 database

☑ Configure a web server for directory server administration

Figure 6-1 ldapxcfg utility

2. Ensure all three check boxes are marked so as to configure each of the items, and click **Next**.

3. Enter the administrator's distinguished name and password similar to that shown in Figure 6-2. The Administrator DN may be any suitable name. Click **Next** to continue.



IBM Directory Server Configuration

Administrator Configuration

Enter a new directory administrator Distinguished Name and password for host m10df5af. All fields are required.

Administrator DN: cn=wq

Administrator Password: ********

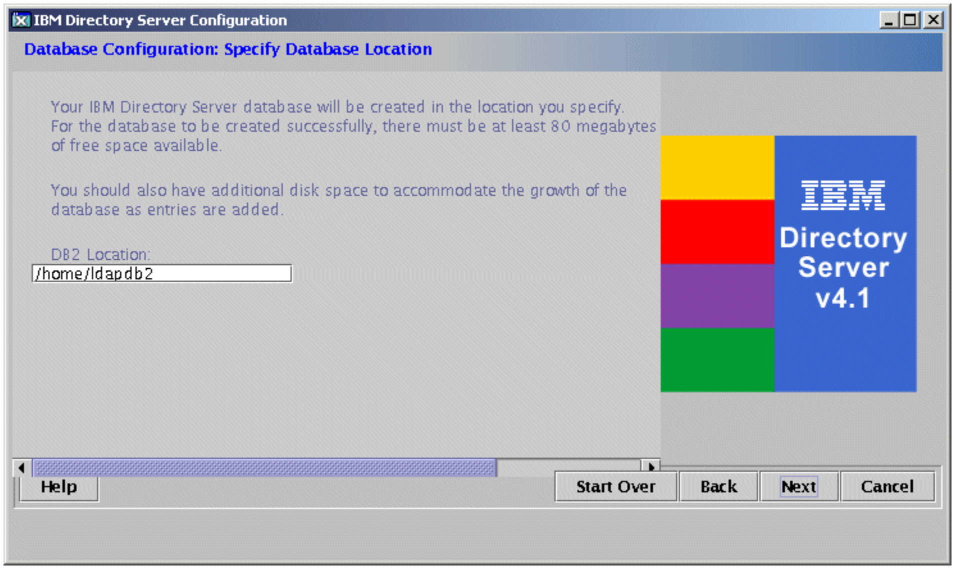
Confirm Password: ********

Figure 6-2 Administrator configuration

4. Select whether the default LDAPDB2 database will be used, or an existing database. The default database will be used here given that it is a new installation. Click **Next** to continue.

5. Select the database character set. This will identify the way in which data is stored. To allow multiple character sets to be stored in the database, **UTF-8** has been selected. Click **Next** to continue.

6. Specify the home directory for the database instance owner. The default is /home/ldapdb2 (as shown in Figure 6-3), or you may choose a suitable alternative. Click **Next** when done.



IBM Directory Server Configuration

Database Configuration: Specify Database Location

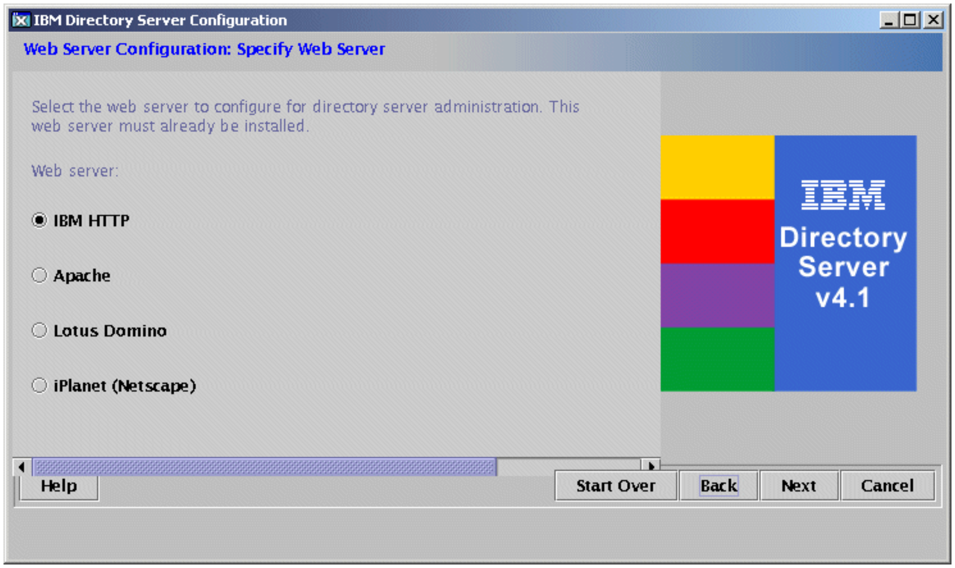
Your IBM Directory Server database will be created in the location you specify. For the database to be created successfully, there must be at least 80 megabytes of free space available.

You should also have additional disk space to accommodate the growth of the database as entries are added.

DB2 Location: /home/ldapdb2

Figure 6-3 LDAPDB2 home directory

7. Specify the Web server to be used for administration of the directory server. The Web server must be installed prior to completing the Directory Server configuration. Click **Next** to continue.



IBM Directory Server Configuration

Web Server Configuration: Specify Web Server

Select the web server to configure for directory server administration. This web server must already be installed.

Web server:

● IBM HTTP

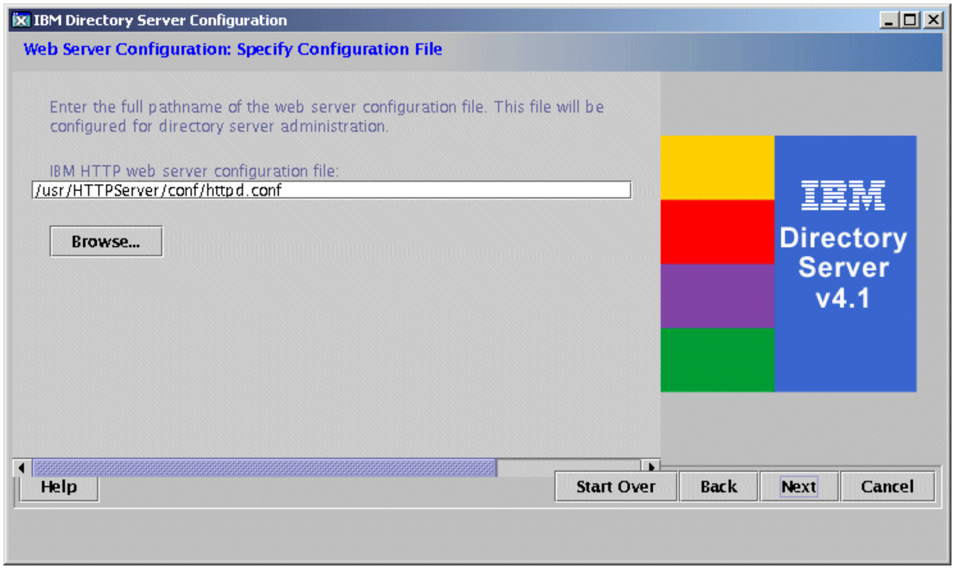
○ Apache

○ Lotus Domino

○ iPlanet (Netscape)

Figure 6-4 Web server selection

8. Specify the location of the Web server's configuration file. Figure 6-5 shows the location of the configuration file for IBM HTTP Server. This will vary according to the type of Web server used, and the installation location. Consult your Web server documentation if unsure. Enter the location in the text box if known, or use the **Browse** button to locate the file. Click **Next** when ready.



IBM Directory Server Configuration

Web Server Configuration: Specify Configuration File

Enter the full pathname of the web server configuration file. This file will be configured for directory server administration.

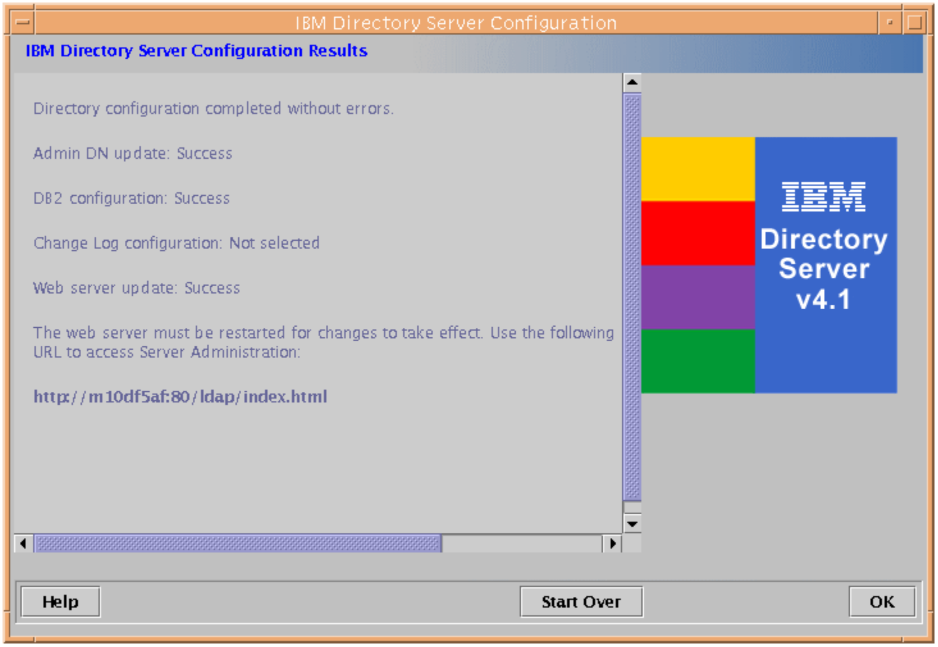
IBM HTTP web server configuration file: /usr/HTTPServer/conf/httpd.conf

Figure 6-5 Web server configuration file

9. Verify that all the setting are correct. Use the **Back** button to change any of the settings prior to continuing. Click **Configure** to apply the changes.

10. On completion, verify the changes were successful by reviewing the displayed results similar to those shown in Figure 6-6. If errors were displayed, click **Start Over** to make the required corrections. Click **Ok** to continue if all was successful.

> **Note:** As the Web servers configuration file was changed, ensure the Web server is restarted following completion of the Directory Server configuration.


IBM Directory Server Configuration
IBM Directory Server Configuration Results
IBM
Directory
Server
v4.1
Help
Start Over
OK

Directory configuration completed without errors.

Admin DN update: Success

DB2 configuration: Success

Change Log configuration: Not selected

Web server update: Success

The web server must be restarted for changes to take effect. Use the following URL to access Server Administration:

http://m10df5af:80/ldap/index.html

Figure 6-6 Configuration results

The initial Directory Server configuration is now complete. So far, the administrator ID has been set, the Directory Server database has been created, and the Web server has been configured to allow administration of the Directory Server through a Web browser. Section 6.1.2, “Starting the IBM Directory Server” on page 70 looks at using the Web browser administration interface in order start the Directory Server, and prepare it prior to adding Connection Manager.

6.1.2 Starting the IBM Directory Server

The IBM Directory Server may be started either from the command line or through the Web interface.

Command line

On UNIX platforms, IBM Directory Server is started by issuing the **slapd** command from a command line. After installation, this file is located in the directory /usr/bin. Typically, /usr/bin will exist in your PATH variable, so the **slapd** command may be run out of any directory. To ensure the Directory Server is running, enter **ps -ef|grep slapd**. The output should look similar to that shown in Example 6-1.

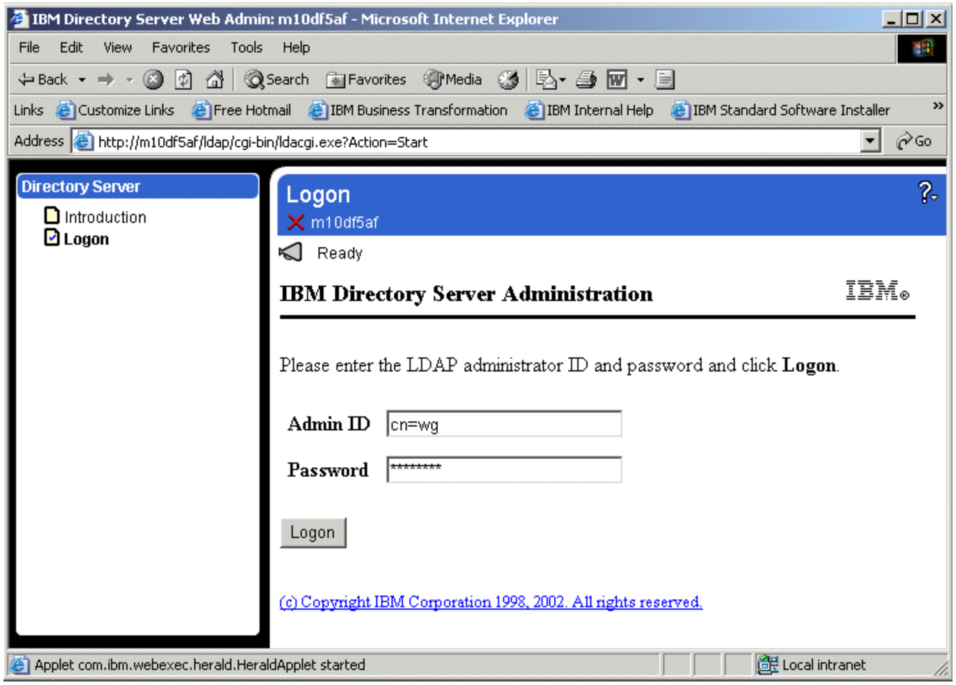
Example 6-1 The slapd process

```
# ps -ef|grep slapd
    ldap 17376     1    0    Jul 11       -   0:01 /bin/slapd
#
```

Web administration

The following steps show how to start the IBM Directory Server using the Web administration tool:

1. Ensure the Web server is running.
2. Start a Web browser, and enter the URL `http://<ldap_hostname>/ldap`.
3. Enter the administrator ID and password as defined in 6.1.1, “Configuring with ldapxcfg” on page 66, and click **Logon**.


IBM Directory Server Web Admin: m10df5af - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Customize Links Free Hotmail IBM Business Transformation IBM Internal Help IBM Standard Software Installer
Address
Go
Directory Server
Introduction
Logon
Logon
m10df5af
Ready
IBM
Logon

Address: http://m10df5af/ldap/cgi-bin/ldacgi.exe?Action=Start

IBM Directory Server Administration

Please enter the LDAP administrator ID and password and click **Logon**.

Admin ID: cn=wg

Password: ********

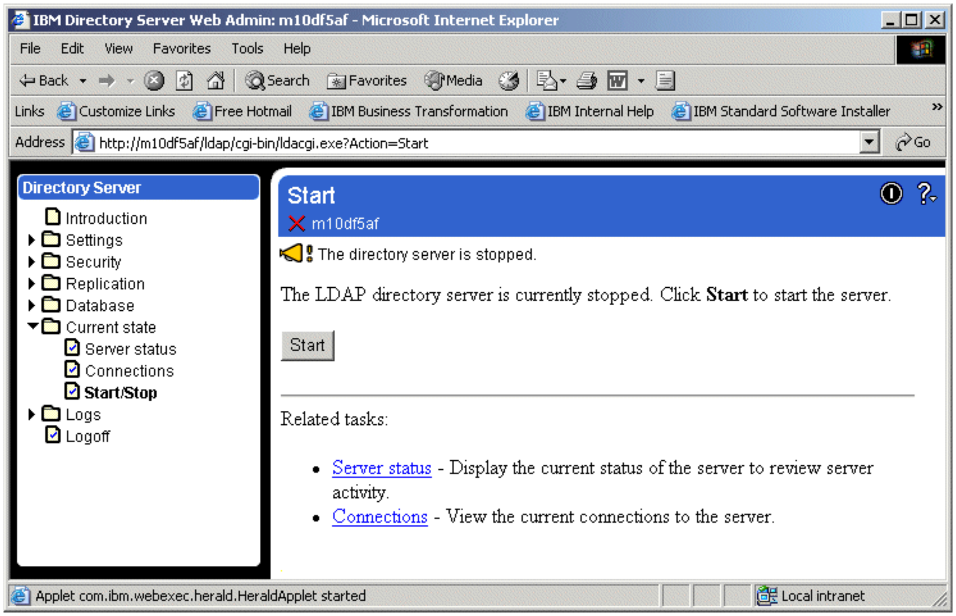
(c) Copyright IBM Corporation 1998, 2002. All rights reserved.

Applet com.ibm.webexec.herald.HeraldApplet started

Local intranet

Figure 6-7 Directory server logon

4. Once logged on, select **Current State -> Start/Stop** from the left hand menu.
5. Click **Start** to start the Directory Server.



The directory server is stopped.

The LDAP directory server is currently stopped. Click **Start** to start the server.

Related tasks:

- Server status - Display the current status of the server to review server activity.
- Connections - View the current connections to the server.

Applet com.ibm.webexec.herald.HeraldApplet started

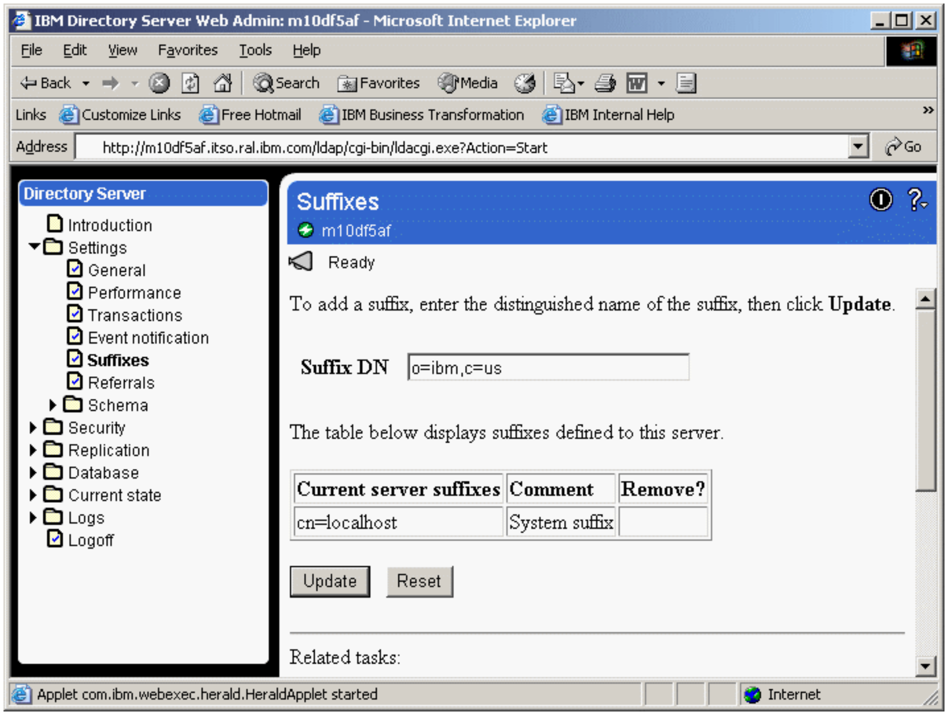
Figure 6-8 Starting IBM Directory Server

6. Check the displayed messages to ensure the Directory Server started without error.

6.1.3 Adding a suffix

Prior to adding Connection Manager, a suffix must be added to the Directory Server. This may be done through the Web administration screens:

1. Log on to the Directory Server administrator as discussed in “Web administration” on page 71.
2. Click **Settings -> Suffixes**
3. Enter the desired suffix in the Suffix DN field shown in Figure 6-9. For the example, we have used `o=ibm,c=us`. Any appropriate suffix may be used according to your own configuration. This suffix will later be identified to the Access Manager as the base distinguished name (DN). Click **Update** when ready.



http://m10df5af.itso.ral.ibm.com/ldap/cgi-bin/ldacgi.exe?Action=Start

To add a suffix, enter the distinguished name of the suffix, then click **Update**.

Suffix DN o=ibm,c=us

The table below displays suffixes defined to this server.

Current server suffixes	Comment	Remove?
cn=localhost	System suffix	

Related tasks:

Applet com.ibm.webexec.herald.HeraldApplet started

Figure 6-9 Adding a suffix

4. After adding a new suffix, the Directory Server must be restarted. This may be done using the **Restart Server** button in the top right of the window.

The Directory Server is now suitably configured to allow us to complete the Connection Manager configuration through the Gatekeeper utility. The first step in this process is to configure the Access Manager. Refer to Chapter 7., “Access Manager configuration” on page 75 for further details.

7

Access Manager configuration

This chapter describes how to configure Access Manager through the Gatekeeper. The Access Manger controls the interaction between persistent data storage, Connection Managers, and Gatekeepers.

This chapter provides details on the following areas:

- Description of Access Manager
- Starting the Gatekeeper
- Configuring Access Manager
- Configuring encryption between Access Manager and the Gatekeeper

7.1 The Access Manager

The Access Manager is a process used to manage communication between the Gatekeeper, persistent data storage, and the Connection Managers. It runs as a daemon called **wgmgrd**, and is automatically installed with the Connection Manager software.

The Access Manager receives and returns commands to and from the Gatekeeper in an XML format. When a command is received from the Gatekeeper, the command is run and the results are converted back into XML for return to the Gatekeeper. If the result of the command creates a change in the configuration, the Connection Manager is automatically notified, and the relevant details are reloaded.

A number of features are available for securing the Access Manager. These are as follows:

- Secure access based on the IP address:

  The Access Manager may be configured to accept connections only from Gatekeepers at specific IP addresses.

- Secure Sockets Layer (SSL):

  The Secure Manager is an optional process that may be installed with the Connection Manager. This enables the use of SSL encryption between the Access Manager and the Gatekeeper, therefore protecting this data flow. The Secure Manager is identified by the **wgmgrsd** process. See 7.2, "Starting the Gatekeeper" on page 77 for more information on creating a secure connection.

- Password encryption:

  By default, all user and administrator passwords are encrypted prior to storing in the Directory Server. If your Directory Server performs its own encryption of passwords, this option should be disabled. This setting must be the same for all Access Managers and Connection Managers that share a common Directory Server. See Figure 7-6 for further details.

- Administrative users:

  You must first log on to the Access Manager using the system's root user ID. Once the initial Access Manager configuration is complete, alternate admin IDs may be created through the Gatekeeper. This allows for an Access Control List to be added to each user, defining the type of access they have to each type of resource. This also eliminates the need to reveal the root password to those only requiring Gatekeeper access.

  Alternatively, you may wish to prevent any remote Gatekeeper connections from being able to log in as root. See Figure 7-8 for additional details.

7.1.1 Access Manager ports

Table 7-1 identifies the default port settings required by Access Manager for communication with the Gatekeeper.

Table 7-1 Gatekeeper ports

Port	Description
9555	Communication between Gatekeeper and Access Manager
9559	Communication between Gatekeeper and Access Manager through SSL

These ports may be changed by altering the relevant setting in the /etc/services file, and then refreshing the **inetd** daemon using one of the following methods:

- AIX

  Execute the command: **refresh inetd -s**
- Linux (using the xinetd daemon)

  Execute the command: **kill -SIGUSR2 `ps -e | grep xinetd | awk '{print $1}'`**.
- Solaris and Linux (using inetd daemon):

  Execute the command: **kill -HUP `ps -e | grep inetd | awk '{print $1}'`**.

7.2 Starting the Gatekeeper

The Gatekeeper is an administration tool used for configuring resources provided by WebSphere Everyplace Connection Manager. On the initial start of the Gatekeeper after installing the software, you will be required to configure the Access Manager. To start the Gatekeeper, do one of the following:

- UNIX: From a command prompt, run the command **wgcfg**
- Windows: Click **Start -> Programs -> IBM Gatekeeper -> Gatekeeper**

The window shown in Figure 7-1 will appear.

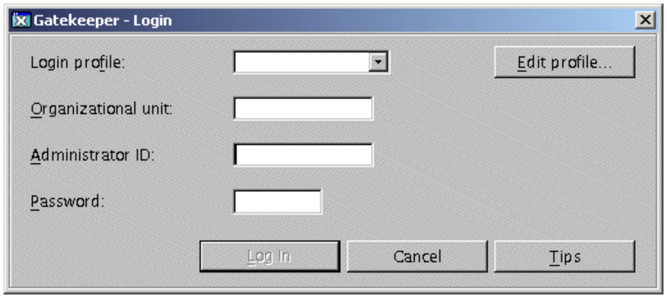
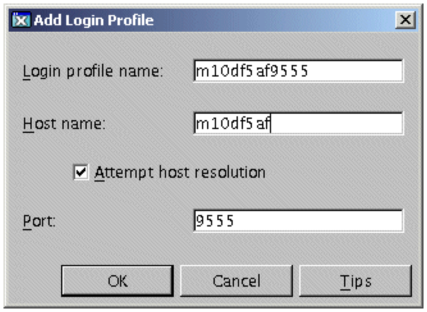


Figure 7-1 Gatekeeper logon

The following steps are used to create a logon profile:

1. Click **Edit Profile...**
2. The communication to the Gatekeeper can either be unsecured or secured using Secure Sockets Layer (SSL):
   - To add an unsecured profile:



Login profile name: m10df5af9555

Host name: m10df5af

☑ Attempt host resolution

Port: 9555

Figure 7-2 Add Logon Profile

   i. Click **Add Profile...**

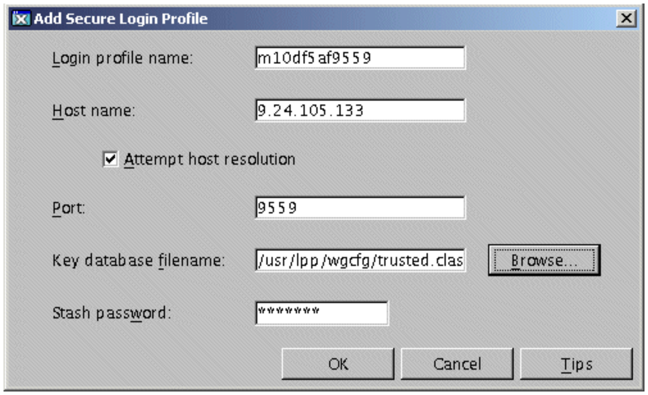
   ii. Add a descriptive name for the Login profile name. Here, we have used the hostname and admin port combination.

   iii. Add the hostname or IP address of the Connection Manager host. If your network does not use a DNS, uncheck Attempt host resolution.

   iv. Either accept the default port 9555, or select an alternate available port. This port is used for communication between the Access Manager and Gatekeeper.

   v. When complete, click **Ok**:

   - To add a secure profile:



Add Secure Login Profile

Login profile name: m10df5af9559

Host name: 9.24.105.133

☑ Attempt host resolution

Port: 9559

Key database filename: /usr/lpp/wgcfg/trusted.clas

Stash password: ********

Figure 7-3 Adding a secure profile

i. Click **Add secure profile...**

ii. Add a descriptive name for the Login profile name.

iii. Add the hostname or IP address of the Connection Manager host. If your network does not use a DNS, uncheck Attempt host resolution.

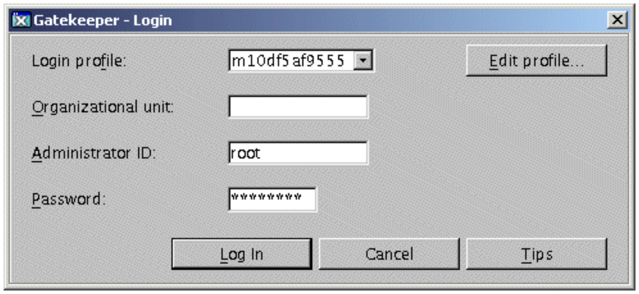
iv. Either accept the default port 9559, or select an alternate available port. This port is used for secure communication between the Access Manager and Gatekeeper.

v. Enter the path and filename of the Java class certificate you will be using. Use **Browse** to locate the file if required.

vi. Enter the password in the Stash password field.

vii. Click **Ok.**

3. To log on, select the required profile from the list, and enter the host's root administrator ID and password to continue. This ID must be used the first time you log on. After creating the Access Manager, you may create additional administrator IDs and roles. Click **Log In** when done.



Gatekeeper - Login

Login profile: m10df5af9555

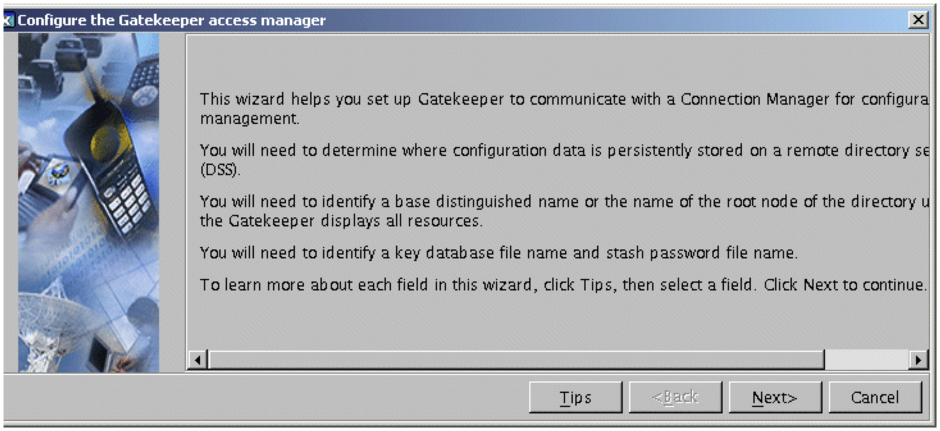
Organizational unit:

Administrator ID: root

Password: *********

Figure 7-4 Gatekeeper logon

4. Click **Accept** to accept the license agreement.
5. The Gatekeeper help window and the Access Manager configuration window now appear. You may close the Help window if not required. Click **Next** on the Configuration window to continue.



This wizard helps you set up Gatekeeper to communicate with a Connection Manager for configura management.

You will need to determine where configuration data is persistently stored on a remote directory se (DSS).

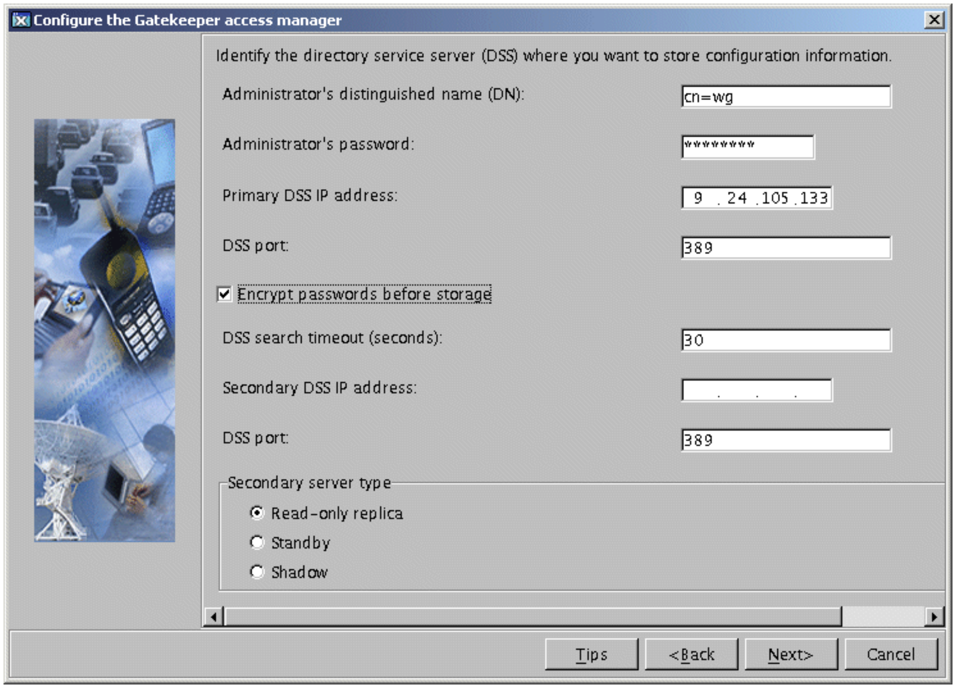
You will need to identify a base distinguished name or the name of the root node of the directory u the Gatekeeper displays all resources.

You will need to identify a key database file name and stash password file name.

To learn more about each field in this wizard, click Tips, then select a field. Click Next to continue.

Figure 7-5 Gatekeeper Access Manager

6. Enter the details of the Directory Service Server (DSS). Use the administrator's the same distinguished name and password as configured in Chapter 6.1, "Configuring the Directory Server" on page 66. If the Directory Server encrypts stored passwords, uncheck Encrypt passwords before storage. Otherwise, leave it checked to allow the Gatekeeper to encrypt the passwords prior to storing it in the Directory Server. If required, add the details of the secondary Directory Server. Click **Next** to continue.


Configure the Gatekeeper access manager

Identify the directory service server (DSS) where you want to store configuration information.

Administrator's distinguished name (DN): cn=wg

Administrator's password: *********

Primary DSS IP address: 9 . 24 . 105 . 133

DSS port: 389

☑ Encrypt passwords before storage

DSS search timeout (seconds): 30

Secondary DSS IP address: . . .

DSS port: 389

Secondary server type

● Read-only replica

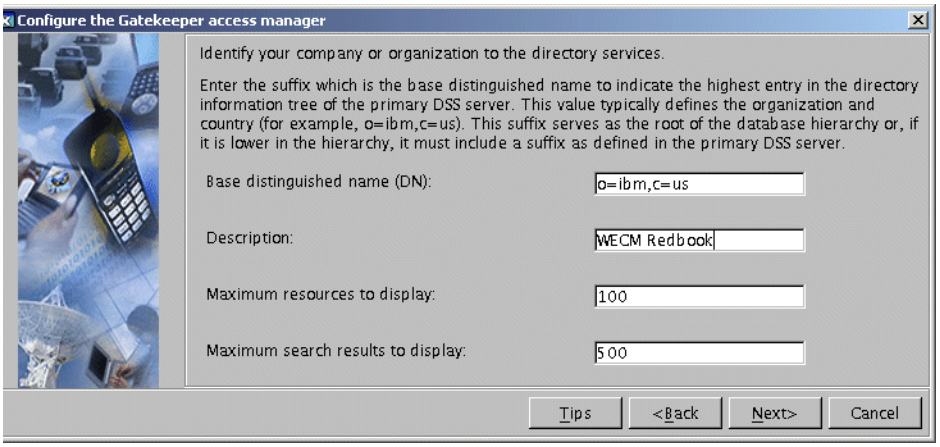
○ Standby

○ Shadow

Figure 7-6 DSS Configuration

> **Tip:** Click **Tips** at any time to get additional information about any of the fields in the window.

7. Add the base distinguished name and an optional description, which will be used to create a directory structure for Connection Manager resources. The base DN should match the root node or suffix added in 6.1.3, “Adding a suffix” on page 72. Alter the maximum display values if required. The actual value used by the Access Manager will be the lesser of the value configured here, and of which configured on the Directory Server. For example, if the Directory Sever is configured to return 50 results from a search, and the value set in the Gatekeeper is 100, only 50 results will be returned. Click **Next** to continue.


Configure the Gatekeeper access manager

Identify your company or organization to the directory services.

Enter the suffix which is the base distinguished name to indicate the highest entry in the directory information tree of the primary DSS server. This value typically defines the organization and country (for example, o=ibm,c=us). This suffix serves as the root of the database hierarchy or, if it is lower in the hierarchy, it must include a suffix as defined in the primary DSS server.

Base distinguished name (DN): o=ibm,c=us

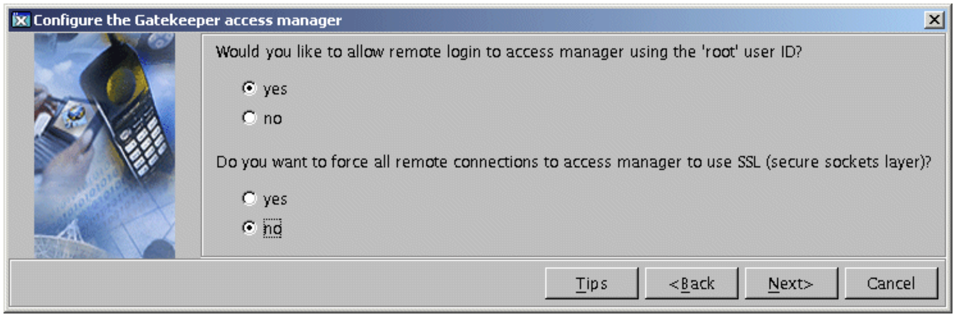
Description: WECM Redbook

Maximum resources to display: 100

Maximum search results to display: 500

Figure 7-7 Additional DSS configuration

8. Select whether or not remote Gatekeeper connections are allowed to log in as the root user, and if all connections to the Gatekeeper require SSL. The default setting of `No` indicates that encryption does not have to be used between the Gatekeeper and Access Manager. If this is set to `Yes`, all log in profiles must be created as secure. Click **Next** to continue.


Configure the Gatekeeper access manager

Would you like to allow remote login to access manager using the 'root' user ID?

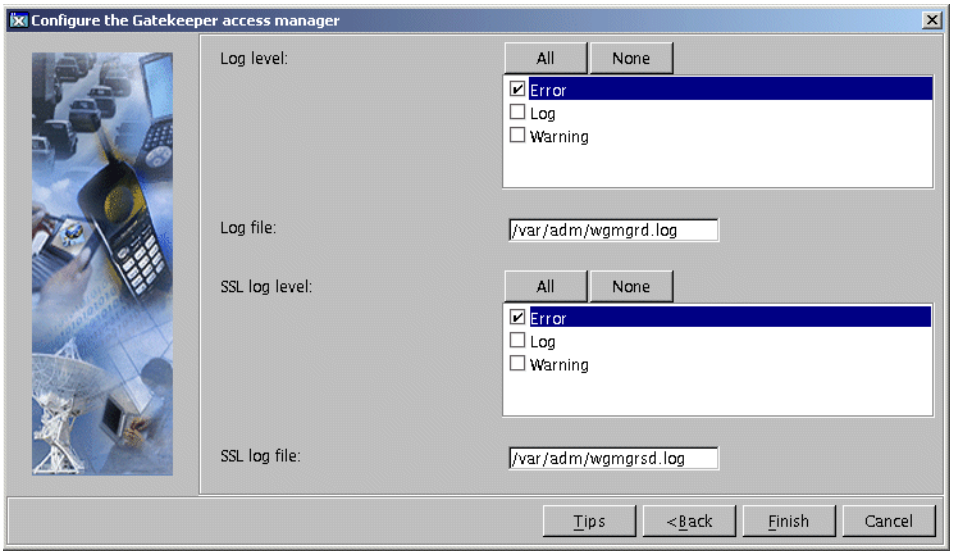
● yes
○ no

Do you want to force all remote connections to access manager to use SSL (secure sockets layer)?

○ yes
● no

Figure 7-8 Root log in and SSL

9. Set the logging level and location for the Access Manager and SSL. In Figure 7-9, only errors will be logged. The location and log level may be modified according to your requirements. Click **Finish** when done.



Log level:

☑ Error
☐ Log
☐ Warning

Log file: /var/adm/wgmgrd.log

SSL log level:

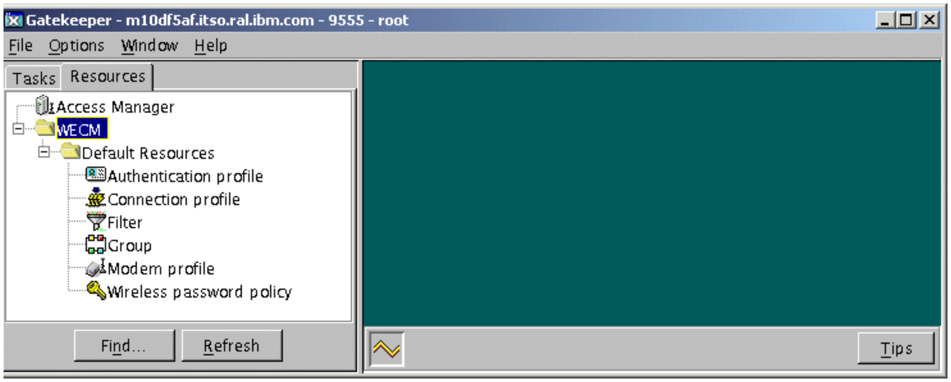
☑ Error
☐ Log
☐ Warning

SSL log file: /var/adm/wgmgrsd.log

Figure 7-9 Log level

The Gatekeeper will now configure the Access Manager as required. This may take some time to complete while the various changes are made to Connection Manager, and updates are made to the Directory Server.

To verify that the configuration was successful, select the **Resources** tab on the Gatekeeper. The view should be similar to Figure 7-10.



```
Access Manager
WECM
  Default Resources
    Authentication profile
    Connection profile
    Filter
    Group
    Modem profile
    Wireless password policy
```

Figure 7-10 Resources

Now that Connection Manager and the Gatekeeper have been installed and configured for use, it is possible to start adding resources. Refer to Chapter 8, “Administration” on page 85 for details regarding the types of resources that may be added, and the steps required to add these.

8

Administration

This chapter describes how to administer WebSphere Everyplace Connection Manager through Gatekeeper.

This chapter also contains information on how to administer users through the administration portlet running in a WebSphere Portal Server environment.

In this chapter, details relating to the following items are provided:

- Review the Gatekeeper Interface
- How to add and manage Connection Manager resources through the Gatekeeper
- Installation of administration portlets
- User administration through the administration portlets

8.1 Administration using Gatekeeper

Gatekeeper is the administrator's console to the Connection Manager. It is stand-alone Java application, and configures and administers the WebSphere Everyplace Connection Manager. Gatekeeper is running on a variety of different platforms (AIX, Windows, Solaris, Linux). It can be located on the same system as the gateway (Connection Manager) or on another system communicating with the gateway and Access Manager through a LAN connection. This communication consists of XML, and can use Secure Sockets Layer (SSL) if a secure connection is required.

You can have many administrators using separate installation of the Gatekeeper at the same time. After you do initial configuration for Connection Manager, you can add administrator IDs for which you must give a different level of access control privileges to administer Connection Manager rather than using root user. When you log in to Gatekeeper, the administrator ID's access control list (ACL) determines which resources you can view and work with. For information about administrator ID and ACL, refer to Chapter 8.1.5, "Using administrator" on page 105.

This section familiarizes you with the ways in which Gatekeeper presents information, and the ways you can use it to administer the resources. Most tasks can be performed in more than one way. For example, to create a resource you can start from the **Tasks** tab and select **Add resource**, or you can right-click an **Organizational Unit** from the **Resources** tab and select **Add resource**.

> **Important:** The Gatekeeper and Connection Manager must be at the same version to operate correctly.

8.1.1 Navigating the Gatekeeper interface

Start Gatekeeper with following command:

- On UNIX platforms, type `wgcfg`
- On Windows platform, select **Start -> Programs -> IBM Gatekeeper -> Gatekeeper**.

After logging on to gateway with your login profile, the Gatekeeper administration window is displayed as in Figure 3-1. For more detailed information about login profiles and the login process, refer to 7.1, "The Access Manager" on page 76.

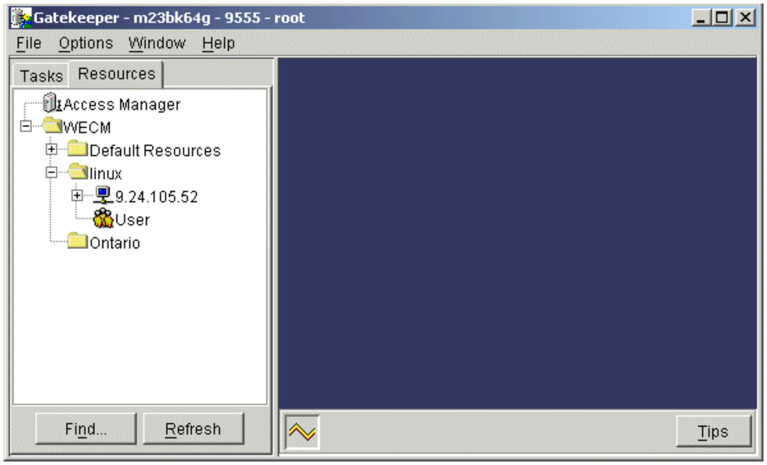


Figure 8-1 Gatekeeper window

The tree structure in the left hand panel shows the Organizational Units (OUs). These containers used to group resources and control access to those resources by administrators. Every resource in a wireless system is assigned to a primary OU. The Gatekeeper provides additional details for all objects by right-clicking over the required items and selecting **What is?** from the displayed menu.

When you double-click some of the resources of each OUs in the left pane, a resulting list of that resource type displays in the right pane. For example, when you double-click **User** in the left pane, you will see a list of users in the right pane. When you see a list of resources in the right pane, you can perform actions on any of the resources in the list.

Each action opens a new window and places it in front of the existing windows. You can arrange the windows in the right pane and bring a window to the front using the Window menu. Right-click a resource to display a context menu with actions appropriate to that resource.

For example, for a Connection Manager resource, you can reset log files, start up or shutdown the Connection Manager, and add resources such as mobile access services, messaging services, and Mobile Network Connections (MNCs). From the context menu for a user ID resource, you can perform actions such as lock, reset failed login count, or reset password. When resources are listed in the right pane, some of these options also displayed at the bottom as buttons.

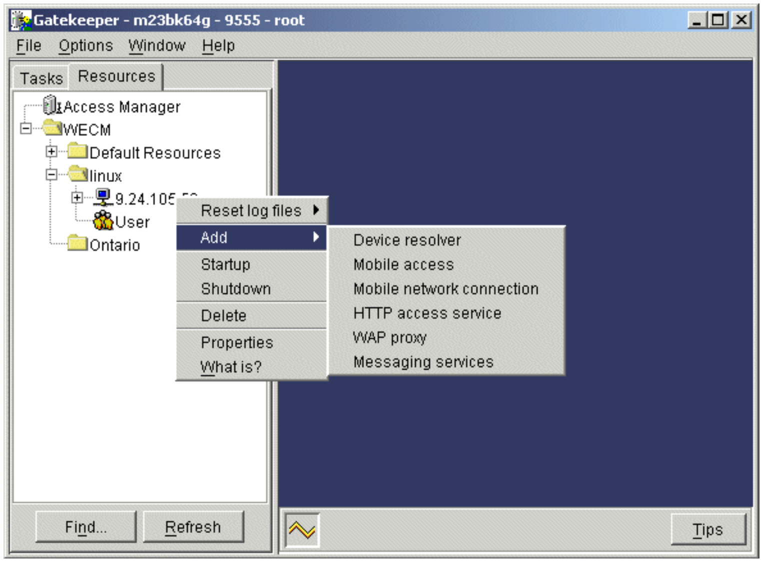
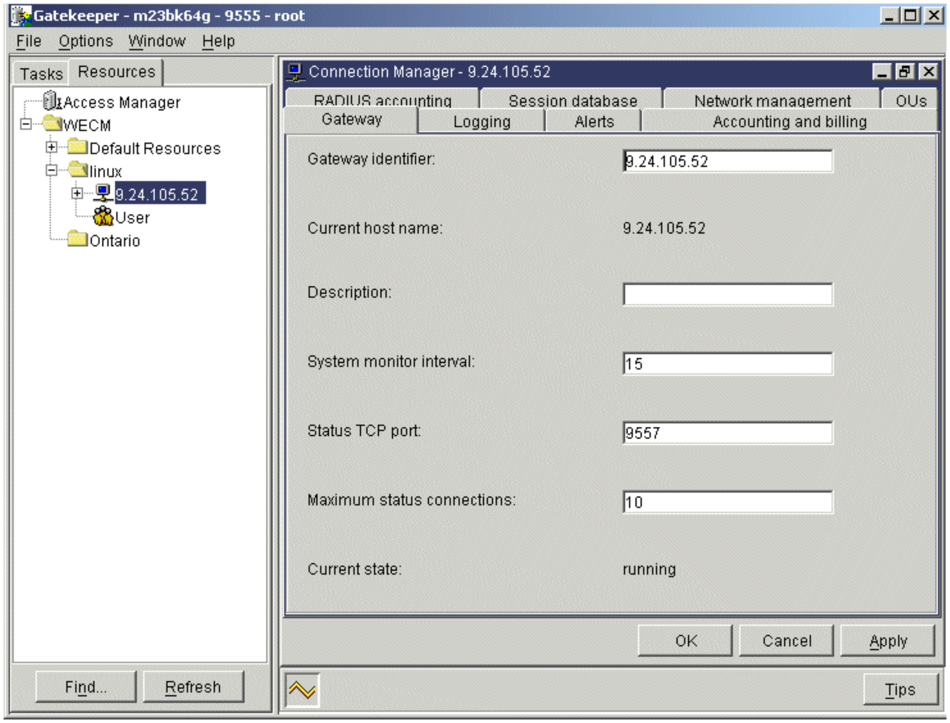


Figure 8-2 Context menu for Connection Manager

Editing resource properties

The Properties window is a tool that lets you view and modify the characteristics of a resource. Properties is an option that appears on the context menu for every type of resource.

To display the Properties window for a resource, right-click the resource in a list and select **Properties**. You can change the values you entered when you created the resource. In some cases, you can enter values that are not required when you create a resource. The properties windows use tabs to group resource parameters. The first tab usually shows the most general parameters including those that identify the resource.



Gateway identifier: 9.24.105.52

Current host name: 9.24.105.52

Description:

System monitor interval: 15

Status TCP port: 9557

Maximum status connections: 10

Current state: running

Figure 8-3 Properties for Connection Manager

Menu bar options

File –> Log In: Starts the Log In dialog, which connects you to the Access Manager. From the Log In dialog, you can create or select the Log In profile you want to use to log into the Access Manager.

File –> Log Off: Disconnects you from the Access Manager to which you were connected, but does not close down the Gatekeeper, enabling you to log in to a different Access Manager.

File –> Change Login Password: Enables you to change the password of the currently logged in administrator ID. This option is not available when the root administrator is logged in.

File –> Login Profiles: Lets you edit the list of profiles that are presented when you log in to the Access Manager.

File –> Access Control Lists: Displays the access control lists of the currently logged in administrator ID.

File –> Exit: Shuts down the Gatekeeper.

The following options control Gatekeeper processes, not the Connection Manager processes:

Options –> Gatekeeper Trace: Tracing is usually turned on only at the direction of service personnel. You can choose:

- The categories of status information from the Gatekeeper subsystems that are written to the trace file MainTrace1.txt.
- To reset the trace file or to append trace information to the trace file at Gatekeeper start-up. By default, each time the Gatekeeper is stopped then started again, the trace file is reset and information is recorded at the top of the file, writing over previous information.
- When the trace file is reset, you can specify whether the old file should be backed up. The next time that the Gatekeeper is started, the previous trace file is renamed to MainTrace1.txt.bak, and includes the current date and time.

> **Note:** The Gatekeeper stores passwords in-the-clear in the trace file. To make sure that passwords are not stored in the trace file, clear the check boxes for the Connection Manager, Communications, and Communications hex dump subsystems.

Options –> Gatekeeper Message Log: You can choose:

- The severity messages, which are written to the message log file, MessageLog.txt
- To reset the message file or to append messages to the file at Gatekeeper start-up. By default, each time the Gatekeeper is stopped then started again, the message file is reset and information is recorded at the top of the file, writing over previous information.
- When the file is reset, whether the old file should be backed up. The next time that the Gatekeeper is started, the previous trace file is renamed to MessageLog.txt.bak, and includes the current date and time.

> **Note:** If the Gatekeeper message log does not correctly display all the characters, choose another operating system tool to view the file.

Options –> Gatekeeper Console Output: After the Gatekeeper trace and message log information has been directed to a file, you can also direct trace and message log information to the console. The console is a window that displays the runtime trace and message information.

Options –> Gatekeeper Properties:

- **Automatic refresh**.

  When a large number of administrators are simultaneously adding and changing resources, the view of the resources may not be updated as quickly as desired. You can choose whether your view of resources is automatically refreshed, and if so, the interval is in seconds. The default value is 60 seconds.

- **Displaying resources**

  Click **Warn me when the list exceeds the maximum setting** to specify that the Gatekeeper displays a message to indicate that the number of resources to display in a list exceeds a configured maximum setting.

  Click **Show find dialog when the list exceeds the maximum setting** to specify that the Find dialog should automatically display, so you can narrow the search for the resources you want displayed.

- **Change Gatekeeper fonts**

  Select the size of the fonts that Gatekeeper displays by sliding the indicator to the desired level.

- **Set the URL or filename of the Information Center**

  Click **Show help menu item for information center** to add **Information Center** to the Help menu. Use this option to launch a Web browser and view the product documentation. To copy the Administrator's Guide PDF file to a Web server to make it available for your organization's administrators and programmers, copy the PDF file of your choice from the installation CD to your Web server's directory. The directory structure on the installation CD is /doc/<lang> where <lang> is your language. And write its URL down to this field.

Window menu: Each time you open a window, its title is added to this menu. You can click the **Window** menu option, then click the title of a window to bring it to the foreground. If you have more than nine windows displayed in the right pane, this menu displays a More Windows option that lists all open windows. You can also select the way you want your open windows displayed within the right pane; they can be cascaded (overlapped diagonally) or tiled (compressed into non-overlapping sub-windows) either vertically or horizontally.

Help -> Help: This menu pops up new window for Help menu. You can choose several forms of help with Gatekeeper.

Help -> Information Center: If you set the URL for information center on the O**ptions -> Gateway Properties** menu, you can see this menu and launch that Administration Guide you have set.

Help -> Product Support Site: This goes to the Web site for Technical Support for WebSphere Everyplace Connection Manager.

Help -> About: This menu also pops up new windows for the Help menu just as **Help -> Help** menu, and on the right pane lets you see the version of your Gatekeeper currently running.

8.1.2 Adding resources

Resources are objects or containers for objects. For example, an OU is a resource that contains other resources such as Connection Managers, users, or mobile devices. When you configure LDAP server for Connection Manager gateway and added Access Manager to it, you will see the primary Organizational Units, Connection Manager on the Gatekeeper. Under the Connection Manager directory, you can see Default Resources, which has the resources come along with the Gatekeeper installation by default.

On the **Task** tab of Gatekeeper, there are resources you can create such as Figure 8-4. You can create each resources by double-clicking it.

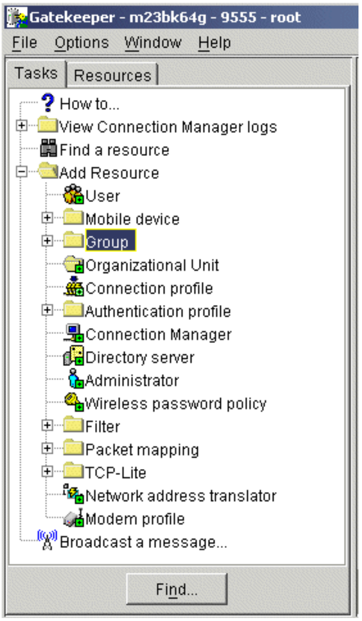


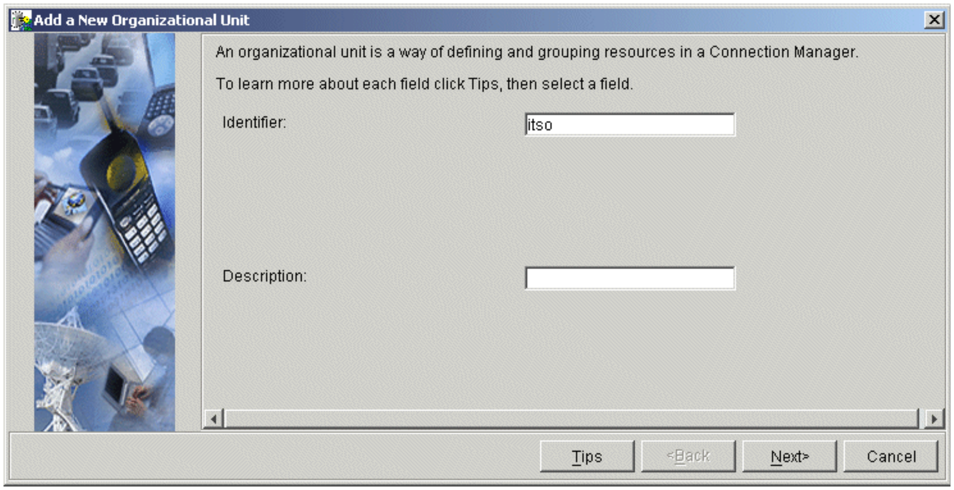
Figure 8-4 Resources on left pane of Gatekeeper

You can start your own management for resources by creating a new Organizational Unit.

Adding Organizational Units

These are the steps:

1. Click the **Resources** tab.
2. Right-click **Connection Manager,** and among context menu select **Add Resource -> Organizational Unit**.
3. Type your Organizational Unit's name. The wizard is shown in Figure 8-5.



Add a New Organizational Unit

An organizational unit is a way of defining and grouping resources in a Connection Manager.

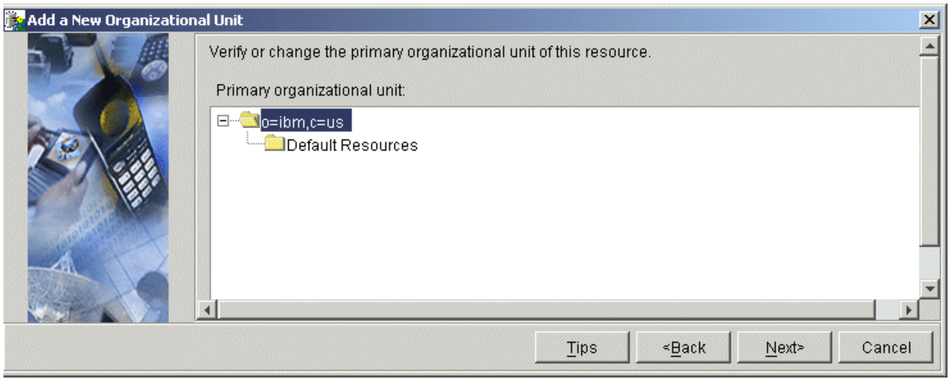
To learn more about each field click Tips, then select a field.

Identifier: itso

Description:

Figure 8-5 Adding new Organizational Unit - step 1

4. Select the primary **Organizational Unit**. This must be `o=ibm,c=us` such as Figure 8-6. Click **Next**.



Add a New Organizational Unit

Verify or change the primary organizational unit of this resource.

Primary organizational unit:

Figure 8-6 Adding new Organizational Unit - step 2

5. You can specify the secondary Organizational Unit if any. Click **Finish**.

Now you can see your new Organizational Unit as in Figure 8-7. You can add resources like Connection Manager on this Organizational Unit. You can add another Organizational Unit under this unit, and manage the various resources efficiently.

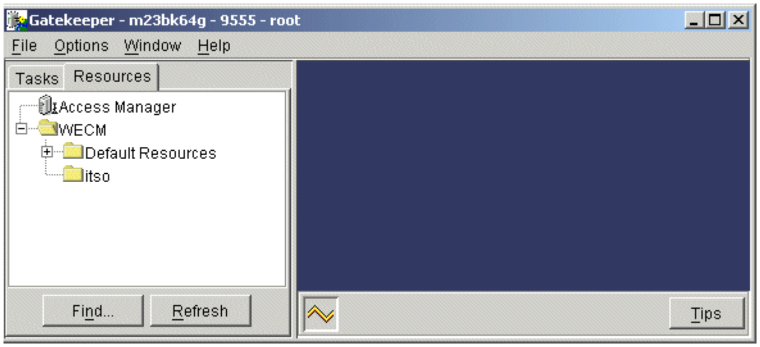


Figure 8-7 Result window after adding Organizational Unit

Like adding an Organizational Unit resource, to add most resources, right-click on the **Organizational Unit** in which the resource will be added, and click **Add Resource**. From the menu, you can select the resource that you want to add. You can also add some resources from selecting resources under the **Add Resource** tree on the **Tasks** tab.

Adding Connection Manager

For each installed WebSphere Everyplace Connection Manager, you need to create a Connection Manager resource on the Organizational Unit (OU).

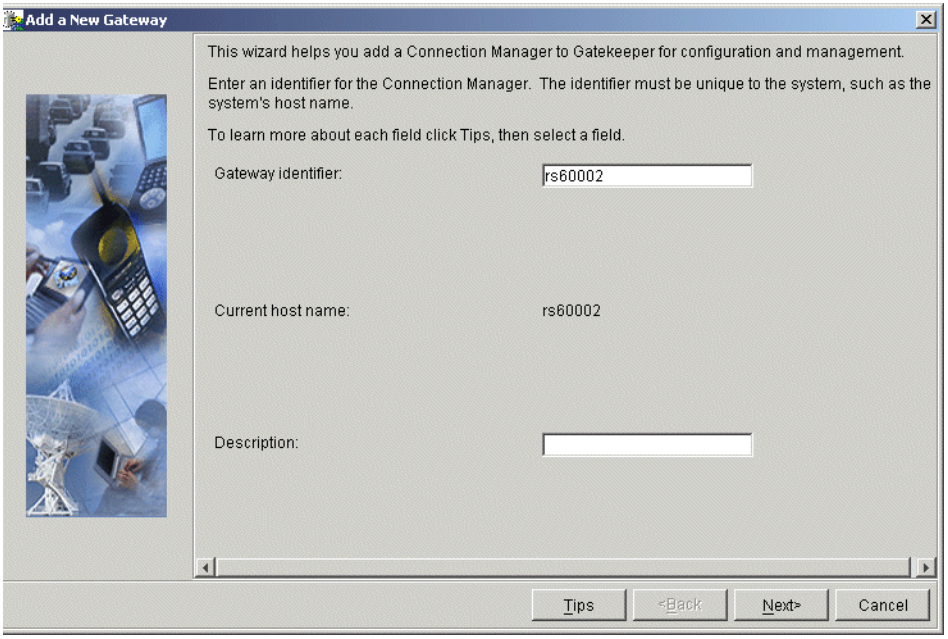
To define Connection Manager to your Organizational Unit, perform following steps:

1. Click the **Resources** tab.
2. Right-click the **OU** in which you want to define Connection Manager.
3. Select Add **Resource -> Connection Manager**

   This will prompt the log-in window to connect to the system in which Connection Manager is installed through its log-in profile.

   You are done with the login profile to your Connection Manager in the installation steps. For more detailed information, refer to 7.2, “Starting the Gatekeeper” on page 77.
4. Select **login profile** and log in to the Connection Manager

   You will see following window such as Figure 8-8.



This wizard helps you add a Connection Manager to Gatekeeper for configuration and management.

Enter an identifier for the Connection Manager. The identifier must be unique to the system, such as the system's host name.

To learn more about each field click Tips, then select a field.

Gateway identifier: rs60002

Current host name: rs60002

Description:

Figure 8-8 Adding a Connection Manager

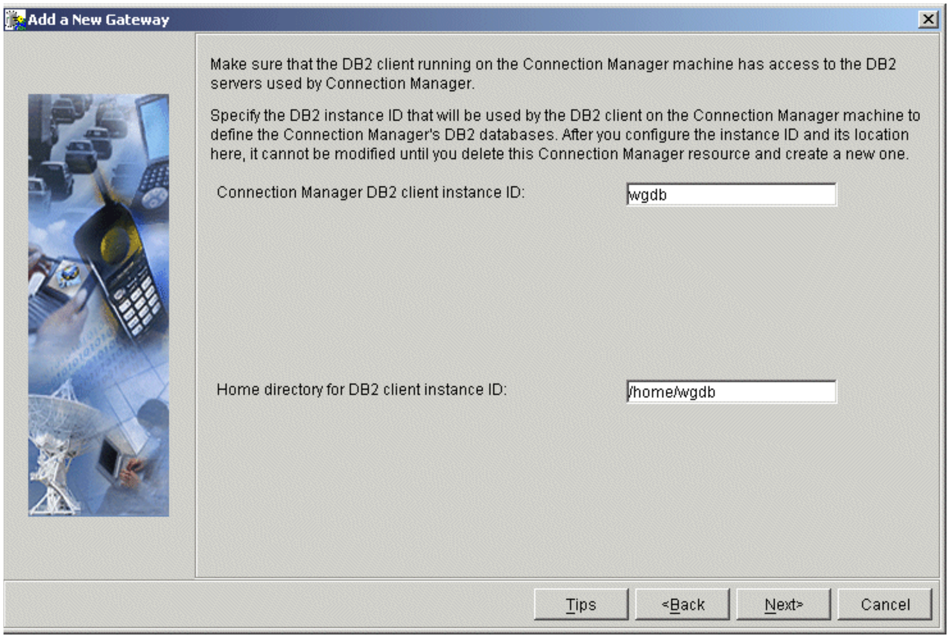
5. Fill in the Identifier for Connection Manager and add a suitable description if desired. Click **Next** when ready.
6. If DB2 UDB or a DB2 client is installed, Connection Manager will automatically detect that this will use DB2 for the Connection Managers persistent storage.

Note: If you are using Oracle Database as your persistent data storage for Connection Manager, assign a database name and a database administrative ID and password. For Oracle, the database name corresponds to the Data Source Name defined in the odbc.ini file.

You need to edit odbc.ini file for Oracle Database to use it as persistent data storage:

1. Edit the odbc.ini file.
2. Add a new line labelled `wgdata` in the section of the file labelled `ODBC Data Sources`. If you installed the accounting and billing support, also add a new line labelled `wgacct` in same section of the file.
3. Use the Oracle8 section as a template and create the corresponding definitions. Update the Description and ServerName key values. ServerName should match one of the entries in your Oracle client tnsnames.ora file.
4. Remove the logon ID and password keys.
5. Save and exit the file.

6. After you click **Next**, you will determine the database instance and home directory for it. In the case where a remote DB2 database is to be used, ensure the following has been completed prior to continuing:
   a. DB2 has been installed on the intended DB2 host.
   b. DB2 client is installed on the Connection Manager host.
   c. Create an instance for the Connection Manager database. When this is created by Connection Manager, the default instance name is `wgdb`. Either use this one or create your own. Connection Manager requires that the instance already exists prior to creating the database for a remote connection.



Make sure that the DB2 client running on the Connection Manager machine has access to the DB2 servers used by Connection Manager.

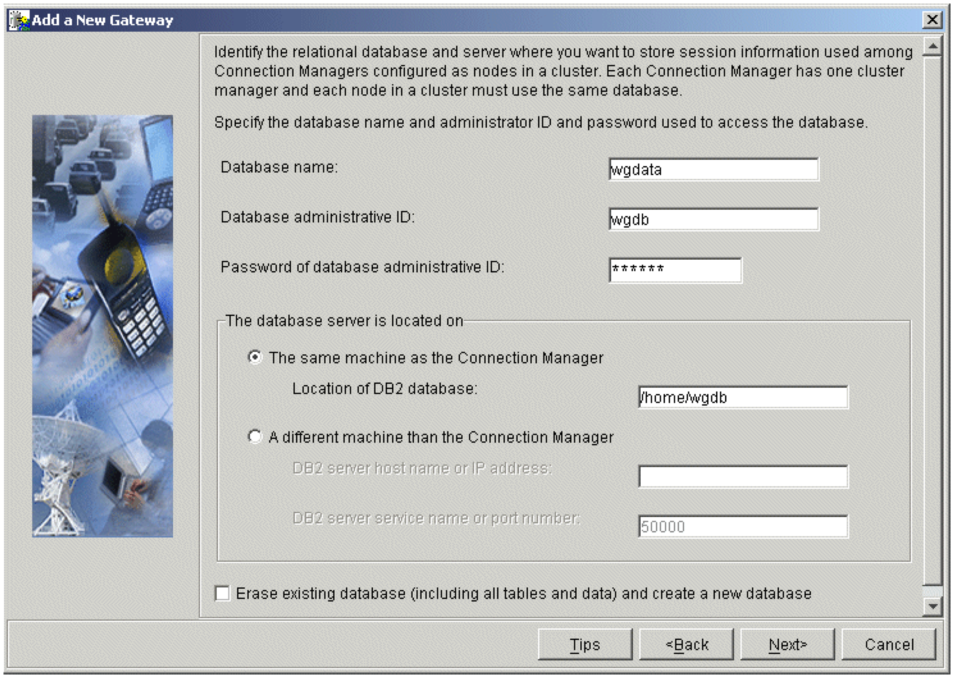
Specify the DB2 instance ID that will be used by the DB2 client on the Connection Manager machine to define the Connection Manager's DB2 databases. After you configure the instance ID and its location here, it cannot be modified until you delete this Connection Manager resource and create a new one.

Connection Manager DB2 client instance ID: wgdb

Home directory for DB2 client instance ID: /home/wgdb

Figure 8-9 DB2 instance details

7. Next, you will choose database name and whether you will use a remote or local database for persistent session data for Connection Manager. Figure 8-10 shows an example of where a local DB2 database is being used. Click **Next** when complete.



Identify the relational database and server where you want to store session information used among Connection Managers configured as nodes in a cluster. Each Connection Manager has one cluster manager and each node in a cluster must use the same database.

Specify the database name and administrator ID and password used to access the database.

Database name: wgdata

Database administrative ID: wgdb

Password of database administrative ID: ******

The database server is located on

● The same machine as the Connection Manager

Location of DB2 database: /home/wgdb

○ A different machine than the Connection Manager

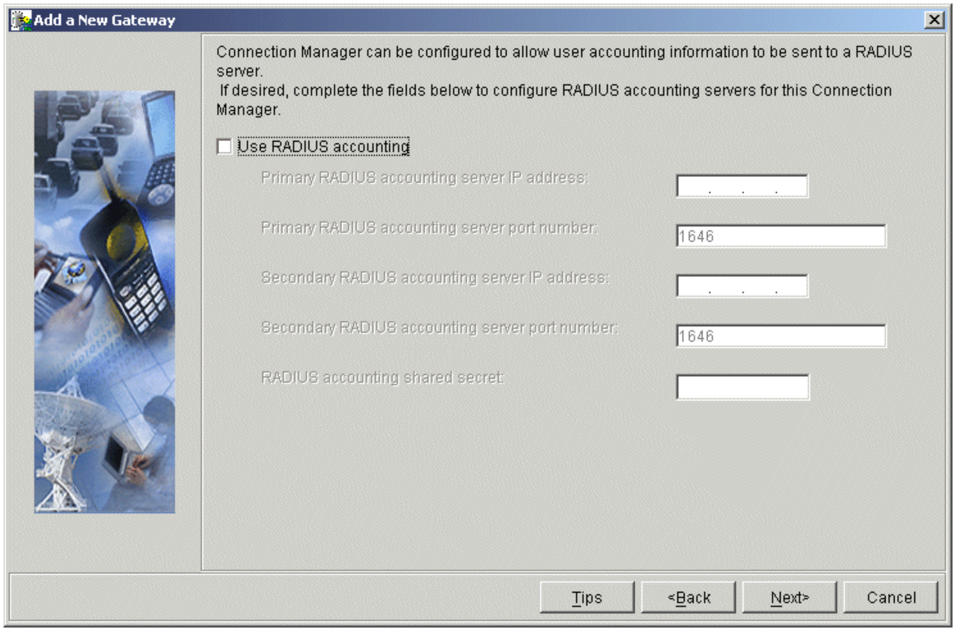
DB2 server host name or IP address:

DB2 server service name or port number: 50000

□ Erase existing database (including all tables and data) and create a new database

Figure 8-10 DB2 database

8. The Connection Manager is able to store accounting and billing records either in a file or in a DB2 database. Select the appropriate option as required, then click **Next**.



Connection Manager can be configured to allow user accounting information to be sent to a RADIUS server.
If desired, complete the fields below to configure RADIUS accounting servers for this Connection Manager.

□ Use RADIUS accounting

Primary RADIUS accounting server IP address:

Primary RADIUS accounting server port number: 1646

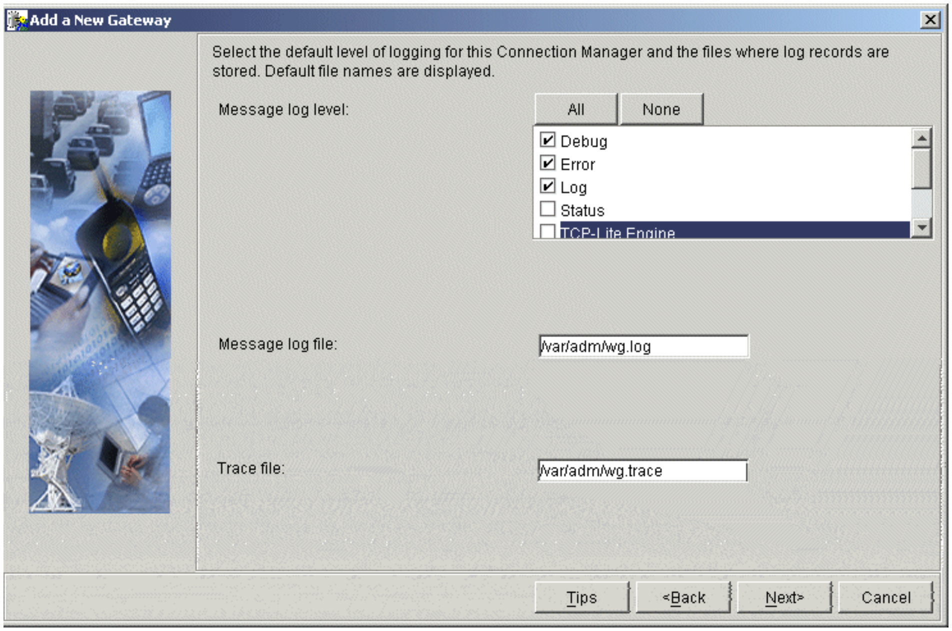
Secondary RADIUS accounting server IP address:

Secondary RADIUS accounting server port number: 1646

RADIUS accounting shared secret:

Figure 8-11 RADIUS accounting options

9. Select the level of logging required for this Connection Manager. With logging set to the highest level, the log files can grow quite large in a short period of time. While the detailed information can be helpful for debugging purpose, logging only errors and warnings is recommended for regular use.



Select the default level of logging for this Connection Manager and the files where log records are stored. Default file names are displayed.

Message log level:

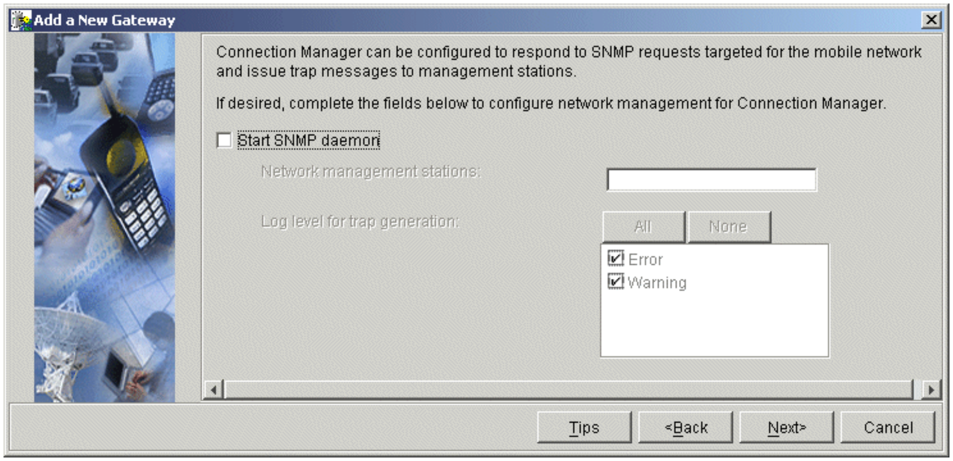
- ☑ Debug
- ☑ Error
- ☑ Log
- ☐ Status
- ☐ TCP-Lite Engine

Message log file: /var/adm/wg.log

Trace file: /var/adm/wg.trace

Figure 8-12 Connection Manager logging

10. Figure 8-13 shows the options for configuring simple network management protocol (SNMP) details. Indicate whether you want the Connection Manager to respond to SNMP requests issued for mobile devices. If you want the Connection Manager to send traps to a Tivoli® NetView® network management station, enter the host name of the station.



Connection Manager can be configured to respond to SNMP requests targeted for the mobile network and issue trap messages to management stations.

If desired, complete the fields below to configure network management for Connection Manager.

☐ Start SNMP daemon

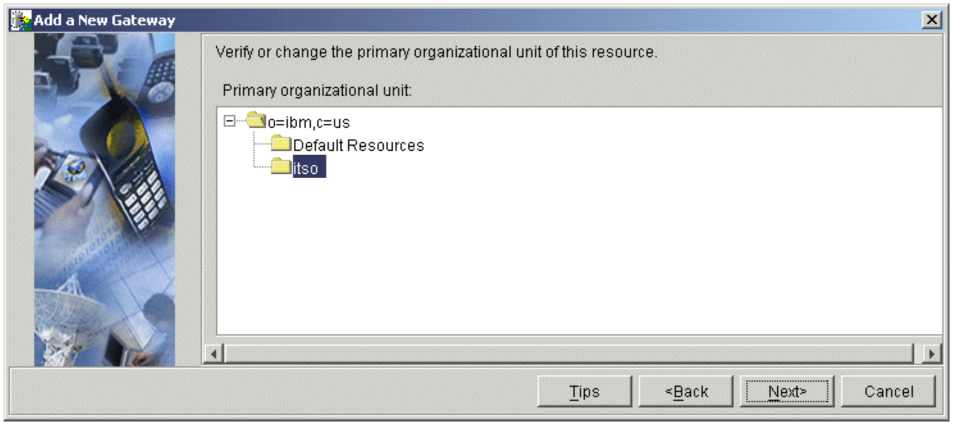
Network management stations:

Log level for trap generation:

- ☑ Error
- ☑ Warning

Figure 8-13 SNMP options

11. Verify the primary Organizational Unit in which this Connection Manager is being created, and in the next step optionally add any secondary OUs.



Add a New Gateway

Verify or change the primary organizational unit of this resource.

Primary organizational unit:

- o=ibm,c=us
  - Default Resources
  - itso

Figure 8-14 Primary OU selection

12. After you click **Finish** Access Manager will create database for persistent session information for Connection Manager, and optionally database for accounting and billing information. Also, this process updates Directory Server (LDAP) for information for Connection Manager.

> **Note:** The process to create a database and update the LDAP server will take some time.

13. After finishing above process, the wizard asks wether you want to add any mobile access services. If you are ready to define your mobile access services, click **Yes**. If you choose not to define it now, you can add it later. See Chapter 9, “Mobile Access Services” on page 137.
14. Again, the wizard asks wether you want to add a WAP proxy. If you are ready to define your WAP proxy, click **Yes**. If you choose not to define it now, you can add it later. See Chapter 19, “WAP gateway” on page 415.
15. The wizard asks wether you want to add a messaging services. Just like mobile services and WAP services, you can add it later. See Chapter 19, “WAP gateway” on page 415.
16. The wizard asks wether you want to add a Mobile Network Connection. You can add it later. See 9.4, “Mobile Network Connection (MNC)” on page 146.
17. Finally, the option is given to start Connection Manager. Click **Yes**. This will start Connection Manager. You can find the process of Connection Manager such as Example 8-1. This may also be started at a later time if preferred.

Example 8-1 Check Connection Manager process

```
[root@m23bk64g bin]# ps -ef |grep wgated
root      5727  5722  0 17:38 ?        00:00:02 /usr/sbin/wgated
root      7156 11669  0 17:49 pts/1    00:00:00 grep wgated
[root@m23bk64g bin]#
```

Adding dependent resources

Some resources are dependent on other resources. For example, mobile access services, messaging services, and MNCs are dependent resources of Connection Manager. WAP services are a dependent resources of a WAP proxy. Application resources are a dependent of messaging services.

These dependent resources are added by right-clicking the parent resource (Connection Manager, WAP proxy, messaging services, etc.) in the left pane of Gatekeeper, and selecting **Add -> dependent resource** on the context menu.

When double-clicked, some resources display lists of those resources in the right pane such as administrators. To add dependent resources, display a list of the parent resources in the right pane, select a specific resource, right-click and select **Add dependent resource**.

8.1.3 Finding resources

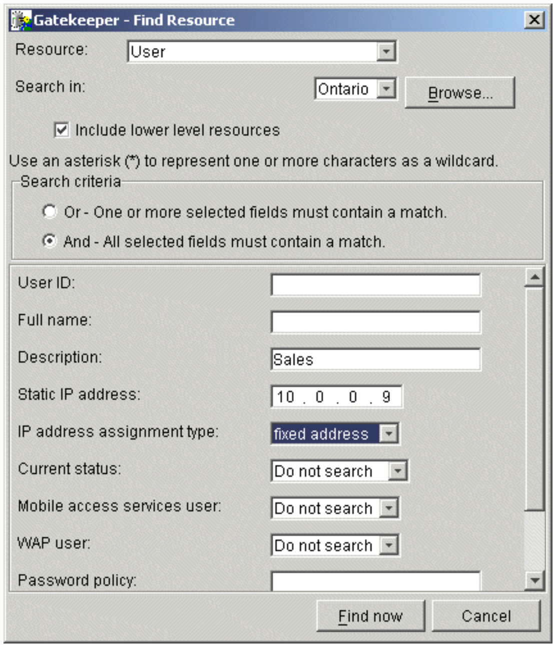
To locate a resource, click **Find** at the bottom of the left pane, or double-click **Find a resource** from the **Task** tab.

Choose a resource from the drop-down list and enter a text string. Choose whether you want to search all OUs or only within a single OU. Use the **Include lower level resources** check box to specify whether to search down the hierarchy below the specified top-level OUs. Click **Find** now to begin the search. Resources matching the search criteria are displayed in a list in the right pane. You can also search for attributes associated with specific resources by specifying the Boolean logical operators `Or` and `And` as search criteria.

For example, if you defined an `OU` as Ontario, and into that `OU` added users with a description of `Sales`. To search for all users in the `OU Ontario` who have fixed IP addresses and who have the description of `Sales` do the following:

1. Click the **User resource**.
2. Click **Browse...** and select the **Ontario OU**.
3. Click **And** in the search criteria field.
4. Click **Fixed Address** in the IP address assignment type field.
5. Enter `Sales` in the Description field.

The window will look like Figure 8-15.



Gatekeeper - Find Resource

Resource: User

Search in: Ontario

☑ Include lower level resources

Use an asterisk (*) to represent one or more characters as a wildcard.

Search criteria

○ Or - One or more selected fields must contain a match.

● And - All selected fields must contain a match.

User ID:

Full name:

Description: Sales

Static IP address: 10 . 0 . 0 . 9

IP address assignment type: fixed address

Current status: Do not search

Mobile access services user: Do not search

WAP user: Do not search

Password policy:

Figure 8-15 Viewing users

6. Click **Find** now.

A wildcard character is a special character that represents one or more other characters. The search supports a wildcard character with the asterisk (*), which represents zero or more characters in a string of characters. For example, entering `john*n` means any word that contains “`john`”, one or more other characters, then a “`n`”. For example, “`johnson`”.

When logged in as the root administrator, you can configure the Access Manager properties of values for the maximum limit of search hits, and the maximum number of resources displayed in a list. To change these values, edit the Access Manager properties. See “Editing resource properties” on page 88.

8.1.4 Viewing Connection Manager logs

Message and trace logs are stored in files on the Connection Manager hosts. The accounting records are stored either in a relational database or in a file on the Connection Manager system. To display these logs from the Gatekeeper, click the **Tasks** tab, then click **View Connection Manager logs**. When you click the log you want to view, wizard panels are displayed on which you can define how you want to view the logs.

Note: To view account, message, or trace logs, the administrator ID must have an additional access enabled by an Access Control Lists with at least read-only access to the Connection Manager. For more information, see "Adding Access Control List (ACL) to administrator ID" on page 109.

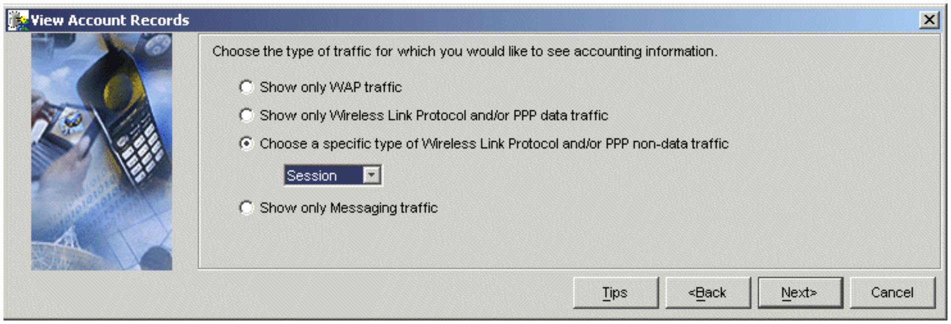
Note: The ability to view message, trace, and account logs is not available on Windows 98 or Windows Me.

You can choose to copy the log from the Connection Manager to the Gatekeeper as a local file. You can also view the most recent entries from the login a dynamic console window. To view recent entries, specify the number of lines from the log or database that you want to display. After the specified number of lines are displayed, the console window is dynamically updated with new entries to the log.

For accounting records, you can additionally filter records to view log entries as shown in Figure 8-16:

- By user ID
- During specific intervals of time
- By type of traffic

When using the Gatekeeper to view accounting records, you may chose to either copy the records to a file, or display them in a real time console depending on your requirements. If they are to be displayed in a console, it is also possible to write the information to a file as it is displayed.



View Account Records

Choose the type of traffic for which you would like to see accounting information.

○ Show only WAP traffic

○ Show only Wireless Link Protocol and/or PPP data traffic

● Choose a specific type of Wireless Link Protocol and/or PPP non-data traffic

Session

○ Show only Messaging traffic

Figure 8-16 Filter accounting records

8.1.5 Using administrator

An administrator is a defined user of the Gatekeeper. The first time you use the Gatekeeper, you must log in as root on the Access Manager system. Using the ID of root is necessary to configure the Access Manager.

After completing the initial configuration of the Access Manager, you may create additional administrator IDs. After you create an administrator ID, in order for it to have access to resources, you must create access control lists (ACL) specifically for that ID.

Each administrator has one or more ACLs, which determine the administrator’s view of wireless resources, and the actions that administrator can perform on the resources in each Organizational Unit.

For example, you can create and administrator ID to work only with user resources, resetting passwords, and locking or unlocking user accounts. Or you can create an administrator ID that can work with any type of resource, but within only one or more specified OUs.

You can specify whether root can log in remotely to the Access Manager using the Gatekeeper. To prevent remote access of administrators using the root ID, edit Access Manager properties on the **Security** tab.

Adding the administrator ID

After logging in to the Access Manager from the Gatekeeper:

1. Right-click the **Organizational Unit** to which you want to add an administrator, and select Add **Resource -> Administrator** such as in Figure 8-17.

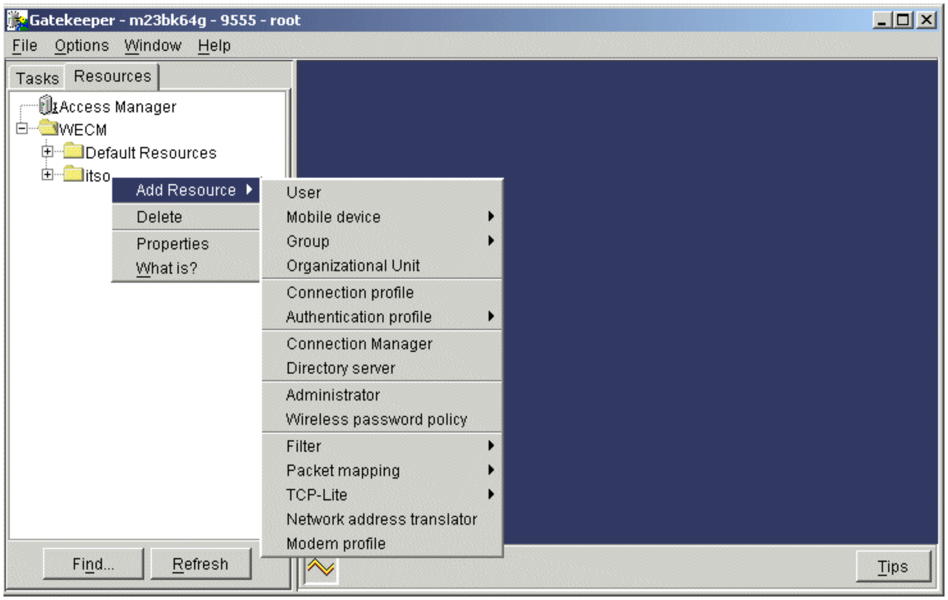
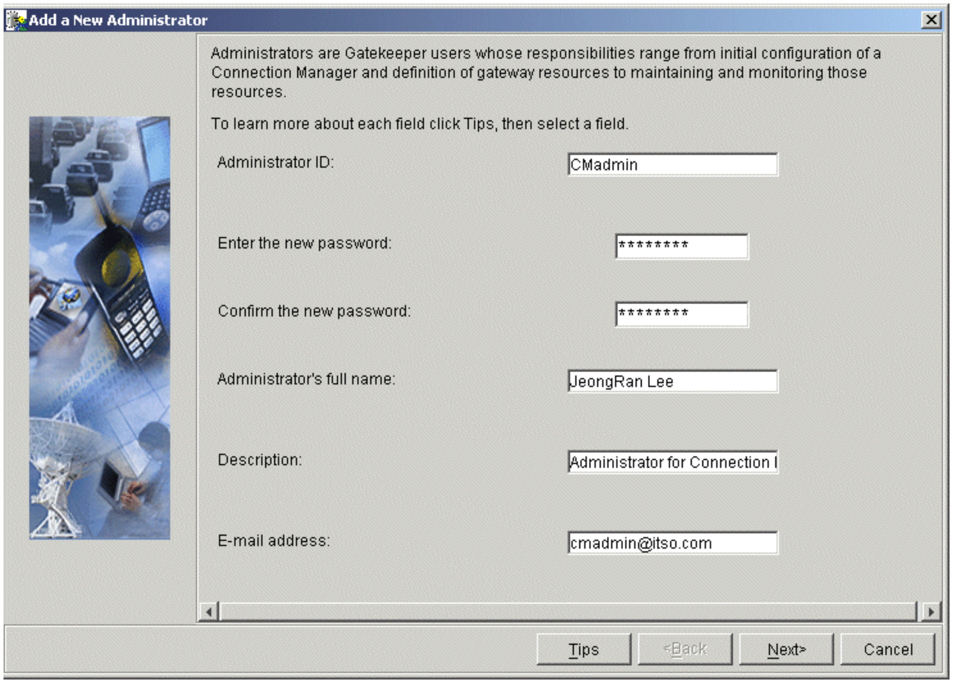


Figure 8-17 Adding an administrator

2. A window as shown in Figure 8-18 will be displayed. Enter the required information for the new administrator similar to the details shown. Click **Next** when complete.



Administrators are Gatekeeper users whose responsibilities range from initial configuration of a Connection Manager and definition of gateway resources to maintaining and monitoring those resources.

To learn more about each field click Tips, then select a field.

Administrator ID: CMadmin

Enter the new password: ********

Confirm the new password: ********

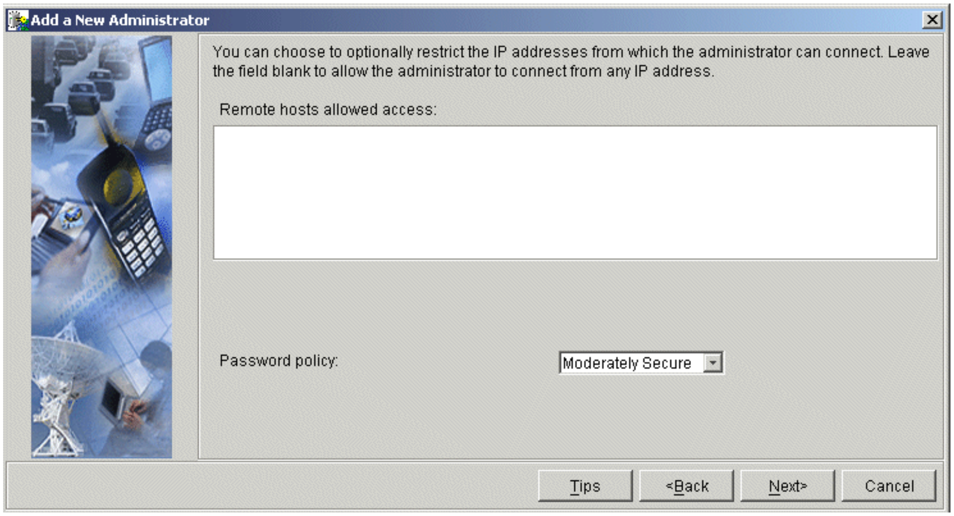
Administrator's full name: JeongRan Lee

Description: Administrator for Connection I

E-mail address: cmadmin@itso.com

Figure 8-18 Administrator details

3. If required, enter an IP address, or a list of IP addresses from which the administrator must connect. Multiple IP address must be separated by a space. If blank, the administrator will be able to connect from any address that has access to the Connection Manager. Click **Next** when done.


Add a New Administrator

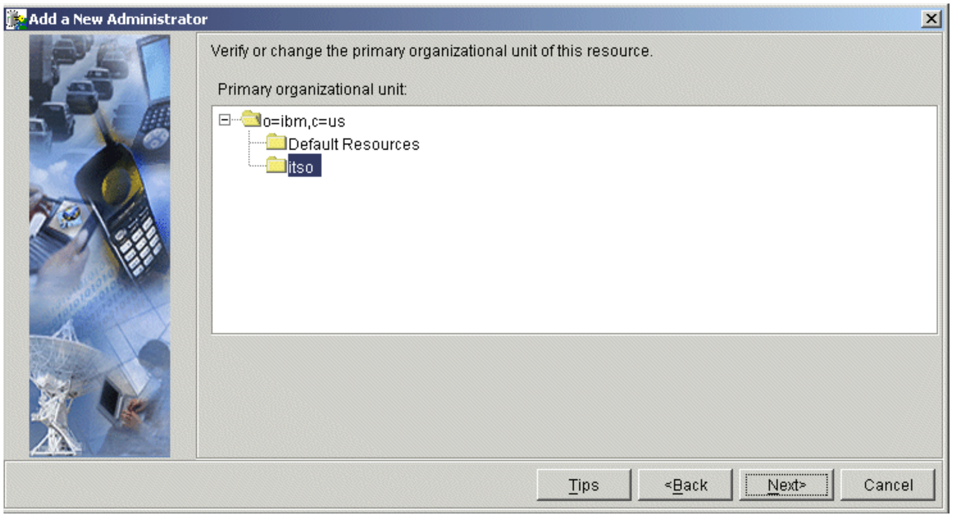
You can choose to optionally restrict the IP addresses from which the administrator can connect. Leave the field blank to allow the administrator to connect from any IP address.

Remote hosts allowed access:

Password policy: Moderately Secure

Figure 8-19 Connection addresses

4. Specify Organizational Unit for this administrator like Figure 8-20 and click **Next**.


Add a New Administrator

Verify or change the primary organizational unit of this resource.

Primary organizational unit:

```
o=ibm,c=us
    Default Resources
    itso
```

Figure 8-20 Primary OU selection

5. Additionally, you may also select a secondary Organizational Unit. Click **Finish** when done.

After adding administrator ID to your Organizational Unit, the wizard will ask whether you will add Access Control List to this administrator ID. You can define the Access Control List later.

Adding Access Control List (ACL) to administrator ID

An Access Control List (ACL) is assigned to an administrator ID and is defined for each resource in an OU to which an administrator has access. Each ACL lists the level of access that the administrator has to the resources in that OU. To view the level of access of the administrator currently logged in to the Gatekeeper, click **File -> Access Control Lists**.

For most resources in an OU, the levels of access available are:

- All

  The highest level of access the administrator can delete and perform all other operations on this resource.

- Create

  The administrator can view, edit, and add resources of this type.

- Modify

  The administrator can view and edit existing resources of this type.

- Read-only

  The administrator can only view resources of this type.

- None

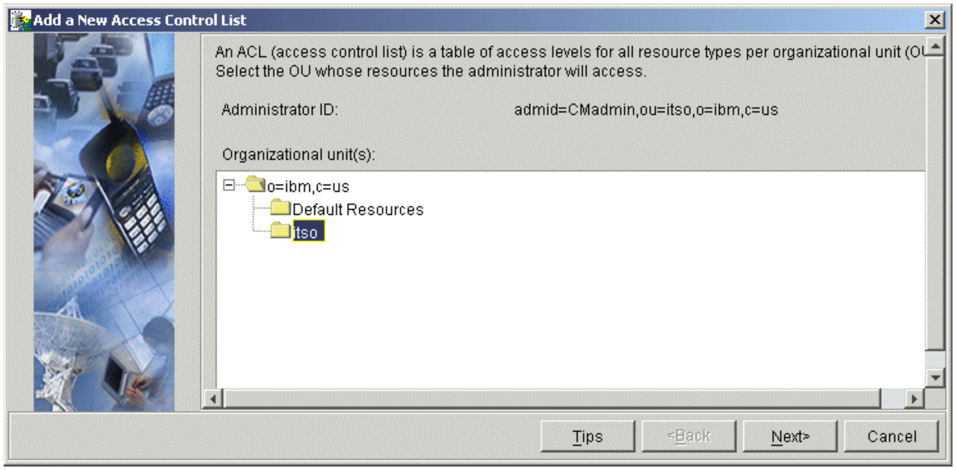
  The administrator cannot view resources of this type.

Each access level includes all those below it. If you assign Create access to a resource in an OU, you also have granted modify and read-only access to the resource in that OU.

Resources that are not directly assigned to an OU inherit access from the parent resource. For example, a cluster manager inherits access from the Connection Manager.

To add ACL to an administrator ID, execute the following steps:

1. After logging in to the Access Manager system through the Gatekeeper, double click **Administrator** on the right pane under your Organizational Unit. You can see the administrator lists on the right pane.
2. Select the **administrator ID** that you want to give an ACL, and click the **Add an ACL** button at the bottom on the right pane. A wizard will pop up like Figure 8-21. Specify the Organizational Unit and click **Next**.



Add a New Access Control List

An ACL (access control list) is a table of access levels for all resource types per organizational unit (OU
Select the OU whose resources the administrator will access.

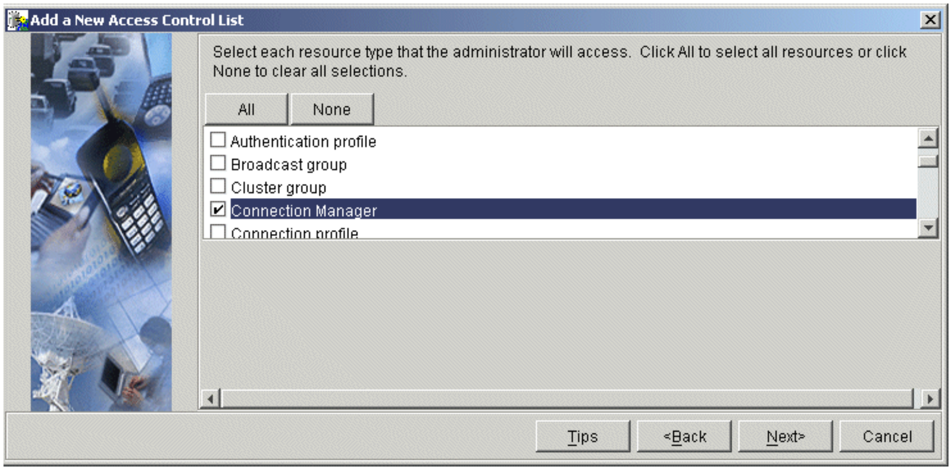
Administrator ID: admid=CMadmin,ou=itso,o=ibm,c=us

Organizational unit(s):

- o=ibm,c=us
  - Default Resources
  - itso

Figure 8-21 Adding ACL - step 1

3. Select the resources type that this administrator will access such as in Figure 8-22. In this example, we selected **Connection Manager resource** for this administrator to control.



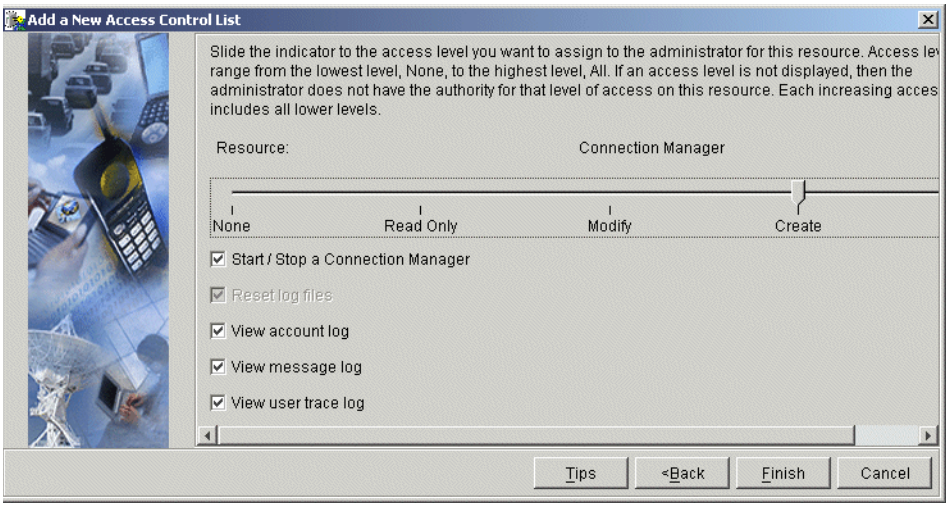
Add a New Access Control List

Select each resource type that the administrator will access. Click All to select all resources or click None to clear all selections.

□ Authentication profile
□ Broadcast group
□ Cluster group
☑ Connection Manager
□ Connection profile

Figure 8-22 Adding ACL - step 2

4. Select the control level of this resource. In Figure 8-23, we selected **Create**. After choosing the privilege, click **Finish**.


Add a New Access Control List

Slide the indicator to the access level you want to assign to the administrator for this resource. Access le range from the lowest level, None, to the highest level, All. If an access level is not displayed, then the administrator does not have the authority for that level of access on this resource. Each increasing acces includes all lower levels.

Resource: Connection Manager

☑ Start / Stop a Connection Manager

☑ Reset log files

☑ View account log

☑ View message log

☑ View user trace log

Figure 8-23 Adding ACL - step 3

You can add another Access Control List later.

In addition to general levels of access, three resources inherit additional levels of authority when assigned specific access levels or above in an OU: Connection Manager, User, and Broadcast group

- Connection Manager access control list:
  - Read-Only:
    - Reset log files: The capability of resetting message, trace, and account log files
    - View account log: The capability of viewing the account log file
    - View message log: The capability of viewing the message log file
    - View user trace log: The capability of viewing the trace log file
  - Modify or Create:
    - Start/stop a Connection Manager: The capability of starting and shutting down Connection Managers
- User access control list
  - Read-only
    - Reset password: The capability of resetting the password of a user ID
    - Reset failed login count: The capability to return a user ID's failed login count to zero

    - Lock/Unlock an account: The capability to toggle a user ID between locked and unlocked
    - Force logoff: The capability to toggle a user ID between locked and unlocked

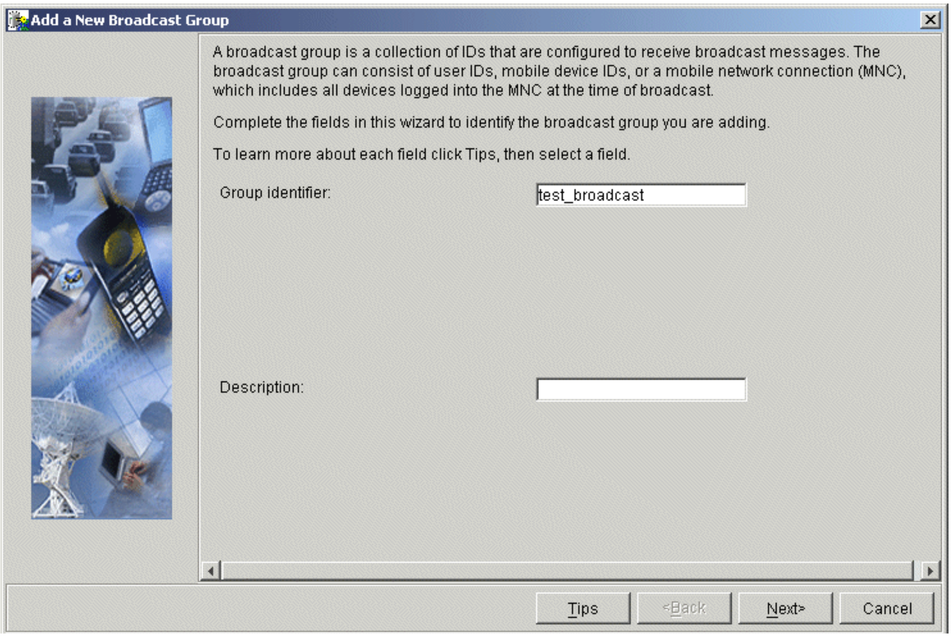
- Broadcast group access control list
  - Read-only
    - Broadcast a message: The capability to send a broadcast message to users

The above access levels are applied to all in upper levels in access control lists by default. For example, the one which is applied to the read-only level is applied to modify and create all levels by default.

8.1.6 Using broadcast groups

You can use the Gatekeeper to define broadcast groups and to issue broadcast messages to these groups. To define a broadcast group:

1. Choose the Organizational Unit (OU) in which you want to create the broadcast group.
2. On the **Resources** tab, right-click that **OU** and select **Add Resource –> Group –> Broadcast group**.


Add a New Broadcast Group

A broadcast group is a collection of IDs that are configured to receive broadcast messages. The broadcast group can consist of user IDs, mobile device IDs, or a mobile network connection (MNC), which includes all devices logged into the MNC at the time of broadcast.

Complete the fields in this wizard to identify the broadcast group you are adding.

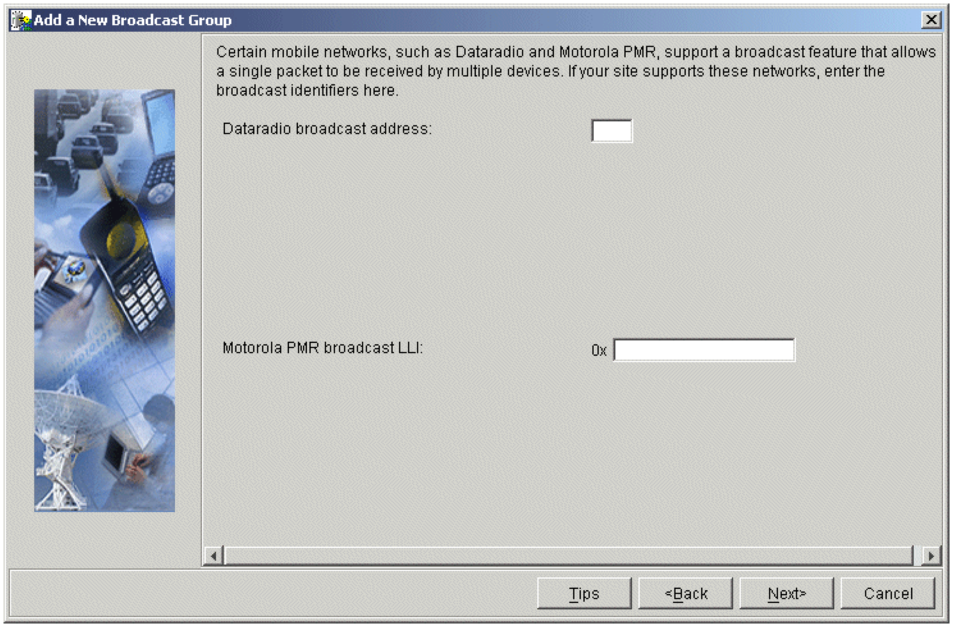
To learn more about each field click Tips, then select a field.

Group identifier: test_broadcast

Description:

Figure 8-24 Adding broadcast group

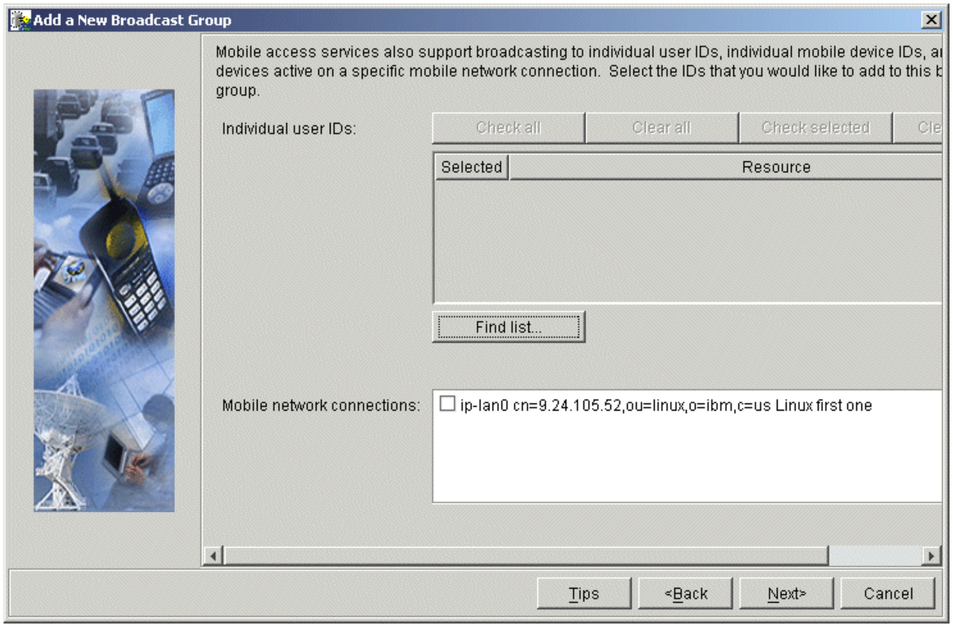
3. You can define a broadcast group for a network with its own broadcast function such as Dataradio or Motorola PMR, or you can create your own group by specifying users and Mobile Network Connections (MNCs).


Certain mobile networks, such as Dataradio and Motorola PMR, support a broadcast feature that allows a single packet to be received by multiple devices. If your site supports these networks, enter the broadcast identifiers here.

Dataradio broadcast address:

Motorola PMR broadcast LLI: 0x

Figure 8-25 Edit DataRadio or Motorola PMR information


Mobile access services also support broadcasting to individual user IDs, individual mobile device IDs, ai devices active on a specific mobile network connection. Select the IDs that you would like to add to this b group.

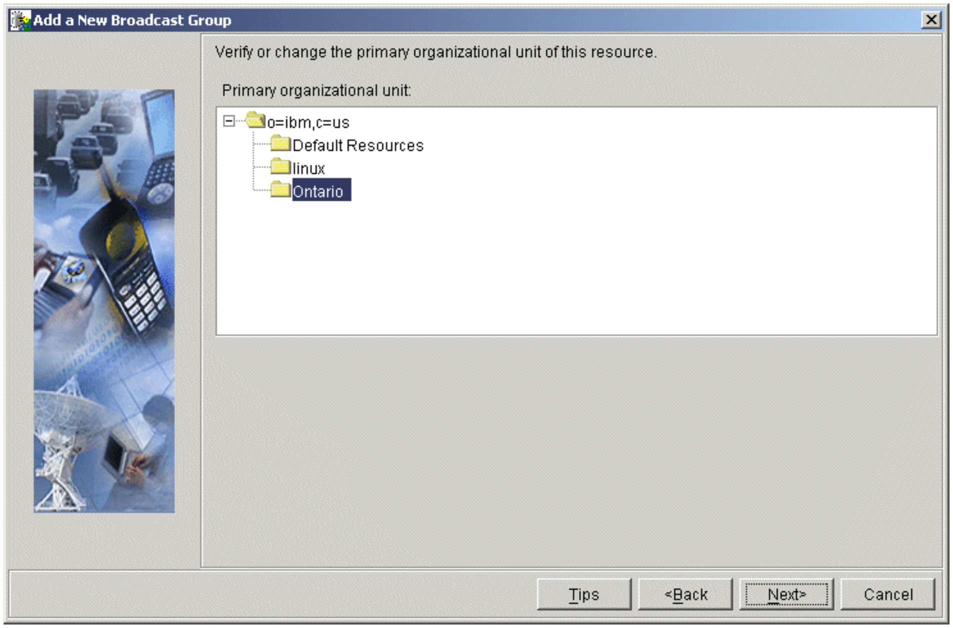
Individual user IDs:

Selected	Resource

Mobile network connections: □ ip-lan0 cn=9.24.105.52,ou=linux,o=ibm,c=us Linux first one

Figure 8-26 Edit users and Mobile Network Connections (MNC) information

4. Verify the primary Organizational Units and optionally select the additional **Organizational Unit** and click **Finish.**



Verify or change the primary organizational unit of this resource.

Primary organizational unit:

- o=ibm,c=us
  - Default Resources
  - linux
  - Ontario

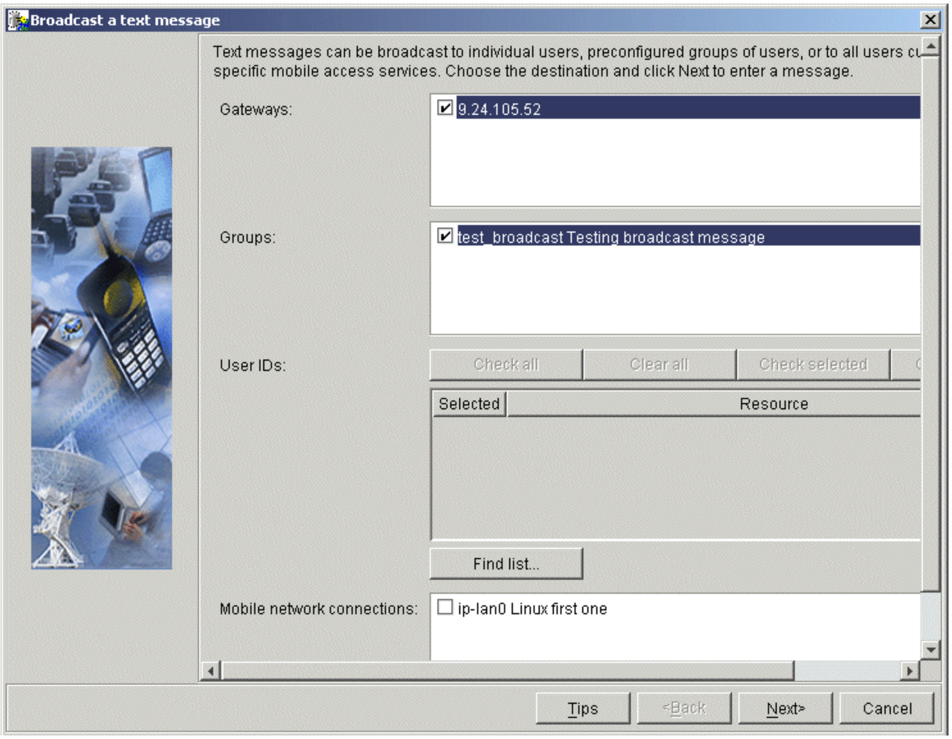
Figure 8-27 Select primary Organizational Unit

After you create a broadcast group, you can change its properties:

1. Double-click the **group** in the Resources tab to list the groups in the right pane.
2. Right-click the **group** in the right pane, then click **Properties**.
3. Modify the desired fields. Use the tabs to access all the group's information.
4. Click **OK** or **Apply**.

After you have created a broadcast group, you can send a broadcast message to its members:

1. Double-click **Broadcast a message** in the **Task** tab.
2. The wizard guides you to select gateways and groups to send a message.



Text messages can be broadcast to individual users, preconfigured groups of users, or to all users cu
specific mobile access services. Choose the destination and click Next to enter a message.

Gateways: ☑ 9.24.105.52

Groups: ☑ test_broadcast Testing broadcast message

User IDs:

Mobile network connections: □ ip-lan0 Linux first one

Figure 8-28 Select gateways, groups, Mobile Network Connections to send a message

3. In the next step, after editing your message, click **Finish**. This will send your message to the selected users.

8.2 Administration portlets

WebSphere Everyplace Connection Manager also provides a portlet interface for user management. This allows WebSphere Portal administrators or administrators of WebSphere Portal based products like WebSphere Everyplace Access to manage both portal and Connection Manager aspects of users. All user management functionality that is offered by Gatekeeper is also available through the portlet interface.

The portlets are not shipped with the installation media but can be downloaded from the IBM Portlet Catalog at:
`http://www.software.ibm.com/wsdd/zones/portal` (NavCode®: 1WP1000EV). The installation of the portlets requires a license key, which can be found on the WebSphere Everyplace Connection Manager CD-ROM.

8.2.1 Prerequisites

Client requirements

The portlets support desktop browser capable of rendering HTML 4.01 content. The browser must support JavaScript 1.5 and inline CSS2. The portlets have been tested on the following browsers:

- Internet Explorer Version 6
- Netscape Navigator Version 6.2
- Opera Version 7
- IBM Home Page Reader Version 3.02.1

Server requirements

- The portlets support WebSphere Portal 4.2.x and all products the are based on this product such as WebSphere Everyplace Access 4.3.
- WebSphere Everyplace Connection Manager 5
- Database server IBM DB2 UDB 7.2 with Fix Pack 7 or Oracle 8.1.7

The portlets should not be deployed on a clustered WebSphere Application Server.

Installation requirements

The installation requires a Java Runtime Environment (JRE) version 1.3.x or higher to be installed on the system on which you are installing the Connection Manager administration portlets. For example, the JRE that is included in WebSphere Application Server can be used.

8.2.2 Installation

WebSphere Everyplace Connection Manager Administration includes two features, which can be installed separately:

- Connection Manager administration portlets
- Connection Manager administration portlet database

This gives you the option to install the Connection Manager Administration portlets in a distributed environment where WebSphere Portal and the database server are running on different systems.

WebSphere Everyplace Connection Manager Administration portlets are provided in a zip file, which contains the installation program and readme files for each supported languages. See the readme file for latest information.

The following steps are being performed during the installation of Connection Manager Administration:

1. Administration portlets are deployed in WebSphere Portal.
2. The place Connection Manager Administration and two pages, Login and Users, are created. The portlets are placed on theses pages.
3. A JDBC Provider named Connection Manager DB Driver and a Datasource named jdbc/wecmadb are created in WebSphere Application Server.
4. A database named WECMADB and a table with the name PROFILES are created an the database server.

If you are installing the Connection Manager Administration portlets feature, make sure that WebSphere Portal is running. If you are installing the Connection Manager Administration portlet database feature, make sure that the database is operational.

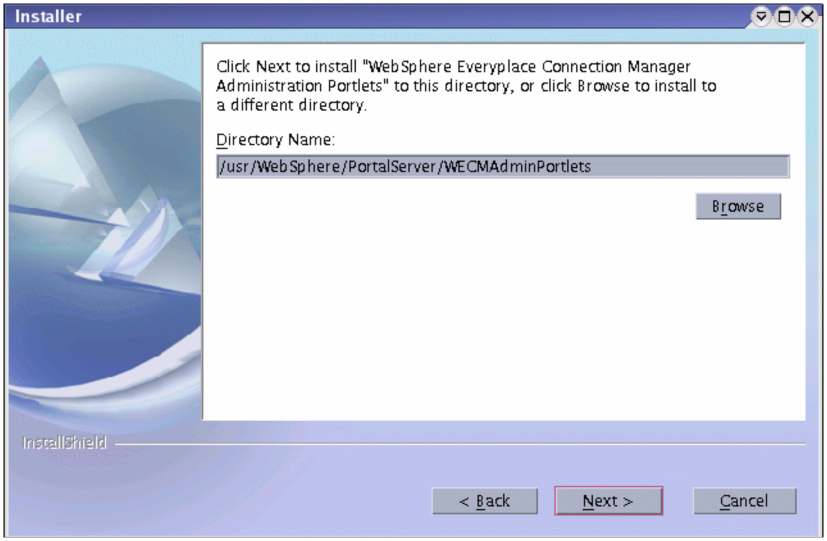
Single-server installation

These are the steps:

1. Extract the contents of the downloaded file to a temporary directory.
2. Open a command line and change to the directory where you extracted the files.
3. Run the following command to start the installation:

   ```
   JAVA_HOME/jre/bin/java -jar WECMAdminPortlets50.jar
   ```

   where `JAVA_HOME` is the root directory of the installed JRE.
4. Click **Next** in the Welcome screen and enter the password in the following screen. The password can be found on the WebSphere Everyplace Connection Manager CD-ROM.
5. Enter the directory in which the portlets will be installed.



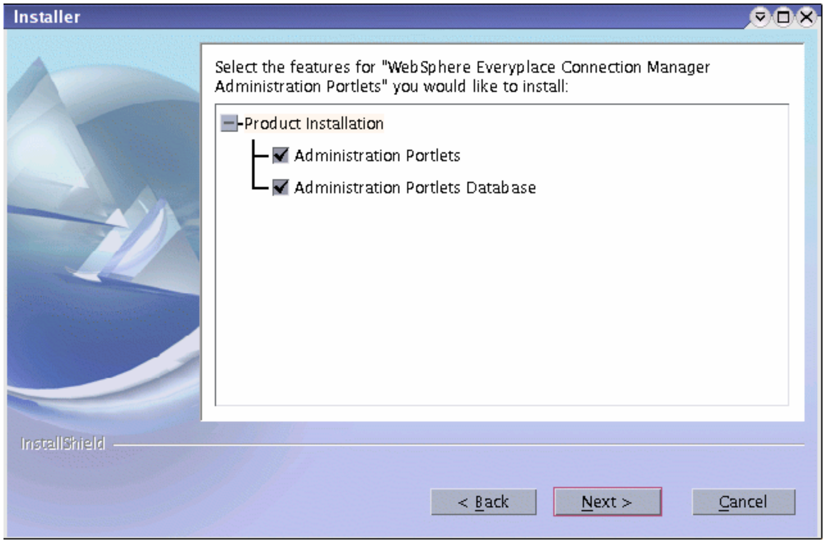
Click Next to install "WebSphere Everyplace Connection Manager Administration Portlets" to this directory, or click Browse to install to a different directory.

Directory Name:

/usr/WebSphere/PortalServer/WECMAdminPortlets

Figure 8-29 Installation directory

6. Select both features, **Administration Portlets** and **Administration Portlets Database**, to install.



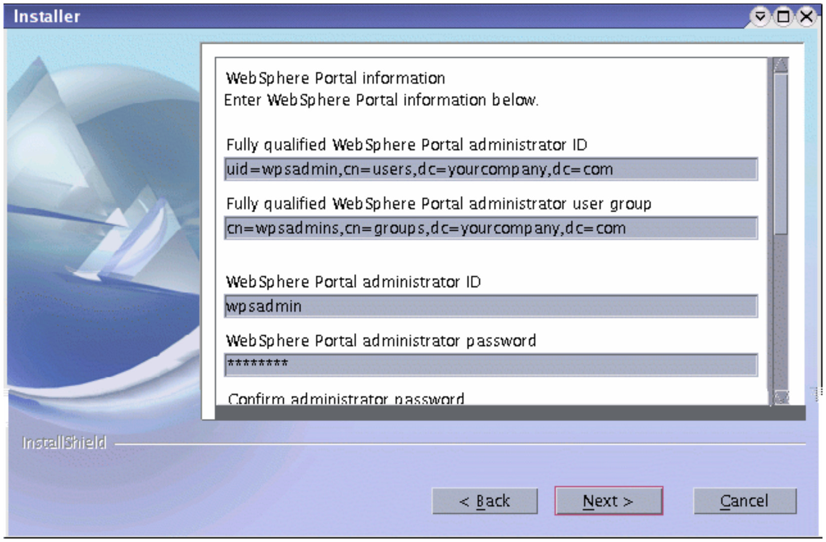
Select the features for "WebSphere Everyplace Connection Manager Administration Portlets" you would like to install:

- Product Installation
  - ☑ Administration Portlets
  - ☑ Administration Portlets Database

Figure 8-30 Feature selection

7. In the next panel, enter the following values:
   - Distinguished name of your WebSphere Portal administrator

- Distinguished name of the WebSphere Portal administrator user group
- WebSphere Portal administrators user ID
- The password of the administrator
- WebSphere Portal hostname
- Base URI of WebSphere Portal
- URI of customized pages
- Installation directory of WebSphere Portal



WebSphere Portal information
Enter WebSphere Portal information below.

Fully qualified WebSphere Portal administrator ID
uid=wpsadmin,cn=users,dc=yourcompany,dc=com

Fully qualified WebSphere Portal administrator user group
cn=wpsadmins,cn=groups,dc=yourcompany,dc=com

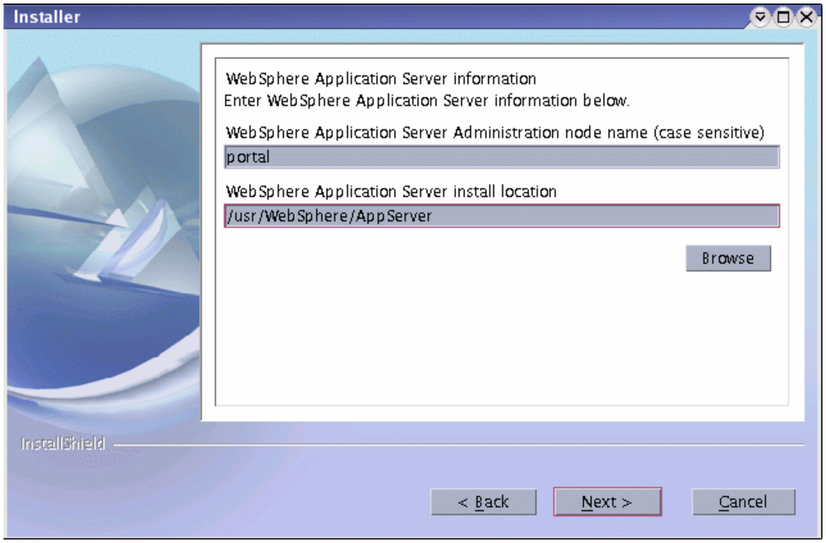
WebSphere Portal administrator ID
wpsadmin

WebSphere Portal administrator password
****

Confirm administrator password

Figure 8-31 WebSphere Portal settings

8. Enter the node name of WebSphere Application Server (WAS) and the installation directory of WebSphere Application Server. The node name is case sensitive and must be entered the same way as it appears in the WebSphere Application Server administration console.



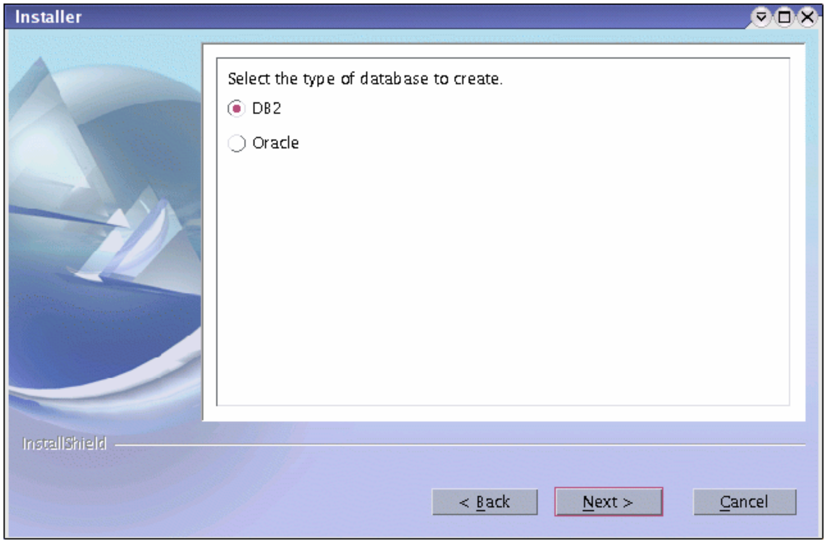
WebSphere Application Server information
Enter WebSphere Application Server information below.

WebSphere Application Server Administration node name (case sensitive)
portal

WebSphere Application Server install location
/usr/WebSphere/AppServer

Figure 8-32 WebSphere Application Server information

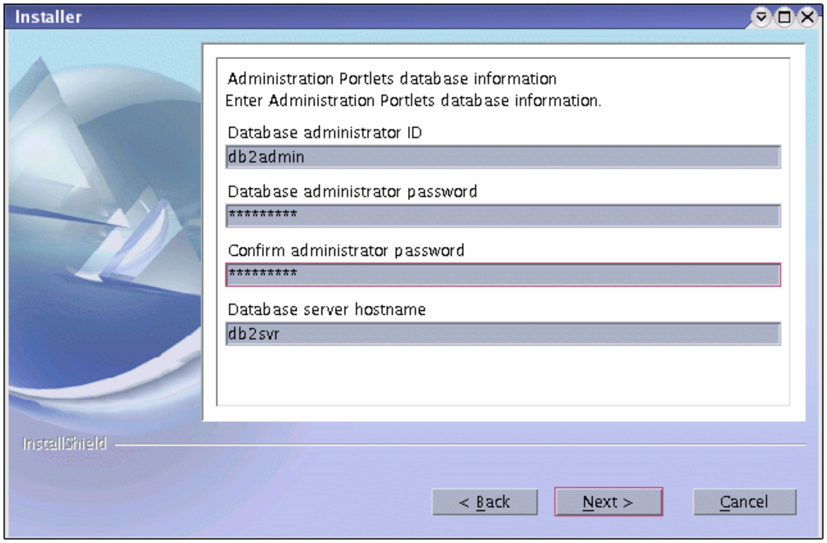
9. Select the type of database, either **DB2** or **Oracle**.



Select the type of database to create.
● DB2
○ Oracle

Figure 8-33 Select database type

10. Enter the information for your database, the database administrator ID, the administrator's password, and the hostname of the database server.



Administration Portlets database information
Enter Administration Portlets database information.

Database administrator ID
db2admin

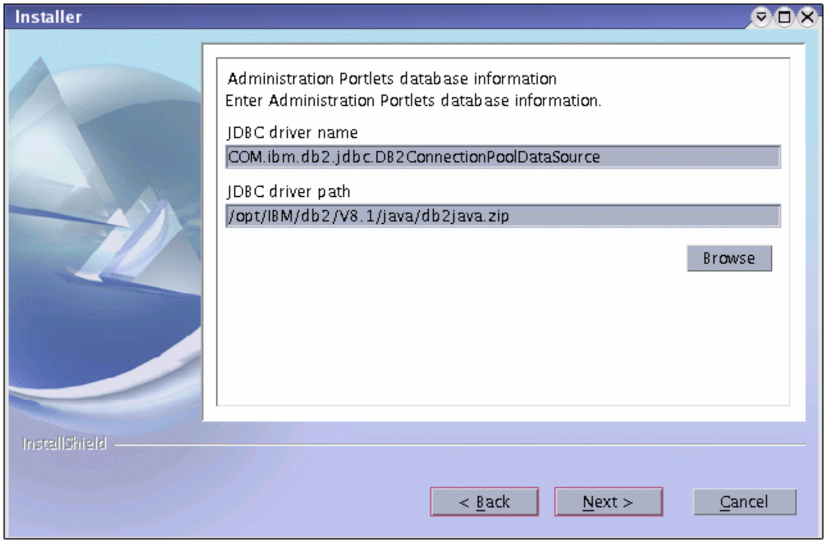
Database administrator password
****

Confirm administrator password
****

Database server hostname
db2svr

Figure 8-34 Database information

11. Enter the values for the JDBC driver: the name of the JDBC driver and its location.



Administration Portlets database information
Enter Administration Portlets database information.

JDBC driver name
COM.ibm.db2.jdbc.DB2ConnectionPoolDataSource

JDBC driver path
/opt/IBM/db2/V8.1/java/db2java.zip

Figure 8-35 JDBC driver

12. The next panel shows a summary of the selected feature. Click **Next** to start the installation.

Installation in a distributed environment

The instructions for the installation in a distributed environment are based on the installation instructions provided in the previous chapter, “Single-server installation” on page 118.

To install the Administration Portlets, perform steps 1 to 8 and 11 on the system where WebSphere Portal is installed. In step 6, select only **Administration portlets**.

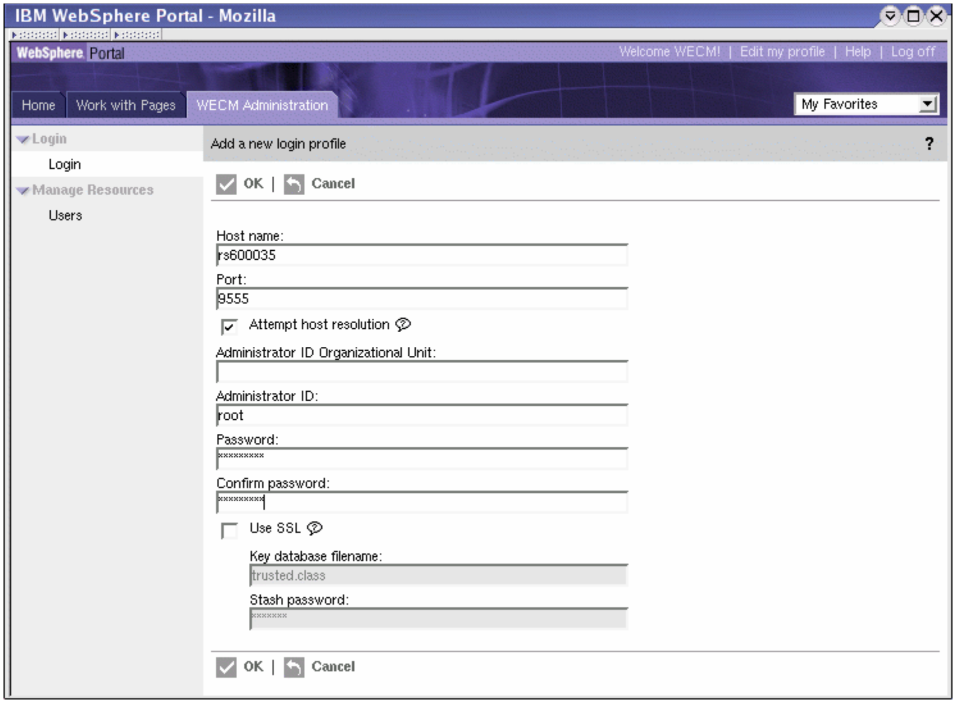
To install the Administration Portlets Database, perform steps 1 to 6 and 9 to 11 on the database server. In step 6, select only **Administration Portlets Database**.

8.2.3 Configuration

Before you use the user management portlets, you have to configure a login profile to the Connection Manager login portlet. A login profile is used to establish a secure connection to a specific Connection Manager. You can add as many login profiles as you want to the login portlet.

To add a new login profile, make sure you are logged in as `wpsadmin`.

1. Select the place **Connection Manager Administration**.
2. Expand the label **Login** and select the page **Login.**
3. Select **Add a login profile.**
4. Fill in all required fields: Host name, Port, Administrator ID, Password, and Confirm Password. Help is available for all input fields. Simply click on the label of the input field.



Host name: rs600035

Port: 9555

☑ Attempt host resolution

Administrator ID Organizational Unit:

Administrator ID: root

Password: *********

Confirm password: *********

□ Use SSL

Key database filename: trusted.class

Stash password: *******

Figure 8-36 Create a login profile

If you use SSL to secure the connection, the portlets ship with the same key database file that is shipped with Gatekeeper. This key database contains the root certificates of various commercial Certificate Authorities such as Verisign, Thawte, and others. If your Connection Manager uses a certificate signed by one of these Certificate Authorities, you can use the key database as is. If your Connection Manager certificate is signed by another CA or uses a self-signed certificate, you have to use the IBM Key Management utility that ships with the Connection Manager and Gatekeeper to create a new database, or update the existing key database. A copy of the key database must be placed on the server where the portlets are installed in the following directory:

```
<was_root>/installedApps/WECMAdmin_WPS_PA_NNN.ear/WECMAdmin.war/WEB-INF/cla
sses
```

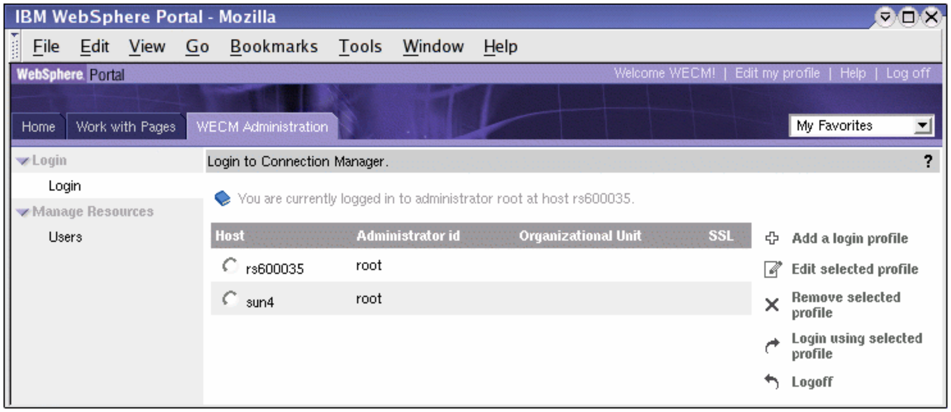
where `NNN` is the portlet application ID that is generated during the installation of the portlets.

4. Select **OK** to create the login profile.

8.2.4 Using the administrative portlets

During installation, one portlet application with two portlets has been installed, and a new place called Connection Manager Administration has been created for the administrator (usually `wpsadmin`). To use the administrative portlets, log in as `wpsadmin` and select the place **Connection Manager Administration**.

Select the login profile you want to work with, and click **Login using selected profile**. The portlet will display a message containing the host and the administrator ID to which you are connected. The message is displayed until you log off. You cannot log in to two Connection Manager at the same time. If you try to log in while you are connected to a Connection Manager, then the current connection will be terminated.



Login to Connection Manager.

You are currently logged in to administrator root at host rs600035.

Host	Administrator id	Organizational Unit	SSL
○ rs600035	root		
○ sun4	root		

Figure 8-37 Login Portlet

Select the **Users** page in the navigation area to go to user management. The first page displayed within the user management portlet is the Search Options page where you can specify different search criteria:

- Organization Units to search

  Select one of the **OUs** listed in the drop-down list in which you want to search, or select **Search all OUs** to search all available Organizational Units.

- Search under this OU

  If you select an individual OU, click the **Search under this OU** check box to select all Organizational Units nested in the selected Organizational Unit's hierarchy.

- Search style

  Select if either all search criteria must match or at least one of the search criteria must match.

- Full name, Description, e-mail address

  Performs a search on the full name, description, or e-mail address respectively. You can use * (asterisk) as a wildcard.

- User ID

  Performs a search on the user ID. The user ID is case-sensitive and must not exceed 32 characters.

- IP address

  Searches for a fixed IP address assigned to the user. The search value must of the format `x.x.x.x` where `x` can be a number between 1 and 255, or * (asterisk) for wildcard searches.

- IP address assignment type

  Can be either DHCP or fixed address

- Current status

  Searches for users based on the status of their connection. This can be either connected, idle, locked, short hold, or WAP.

- Mobile access services user, WAP user

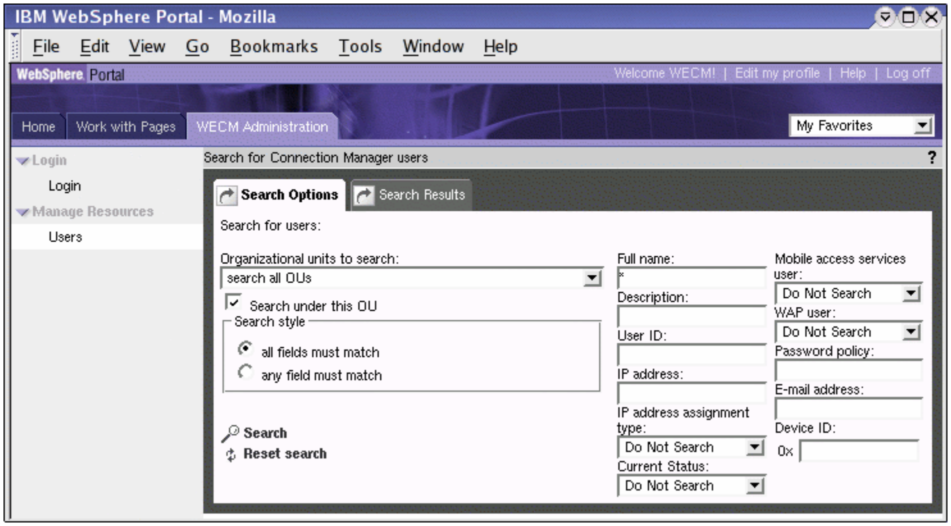
  Select **True** if the search should only return mobile access services users or WAP users, respectively. Select **False** if the search should exclude the mobile access services user or WAP user respectively.

- Password policy

  Searches for the name of a password policy

- Device ID

  Specifies the unique device identifier associated with a given device. This field is initially cleared and after a wireless client logs in using the password key exchange for the first time, the identifier of the device is stored in this field. Alternately, an administrator can fill in the device identifier before the wireless client logs in. The device identifier must be all uppercase hexadecimal digits using no spaces or punctuation.


IBM WebSphere Portal - Mozilla
File Edit View Go Bookmarks Tools Window Help
WebSphere Portal
Welcome WECM! | Edit my profile | Help | Log off
Home
Work with Pages
WECM Administration
My Favorites
Login
Login
Manage Resources
Users
?

Search for Connection Manager users

Search Options | Search Results

Search for users:

Organizational units to search: search all OUs

☑ Search under this OU

Search style

● all fields must match

○ any field must match

Search

Reset search

Full name: *

Description:

User ID:

IP address:

IP address assignment type: Do Not Search

Current Status: Do Not Search

Mobile access services user: Do Not Search

WAP user: Do Not Search

Password policy:

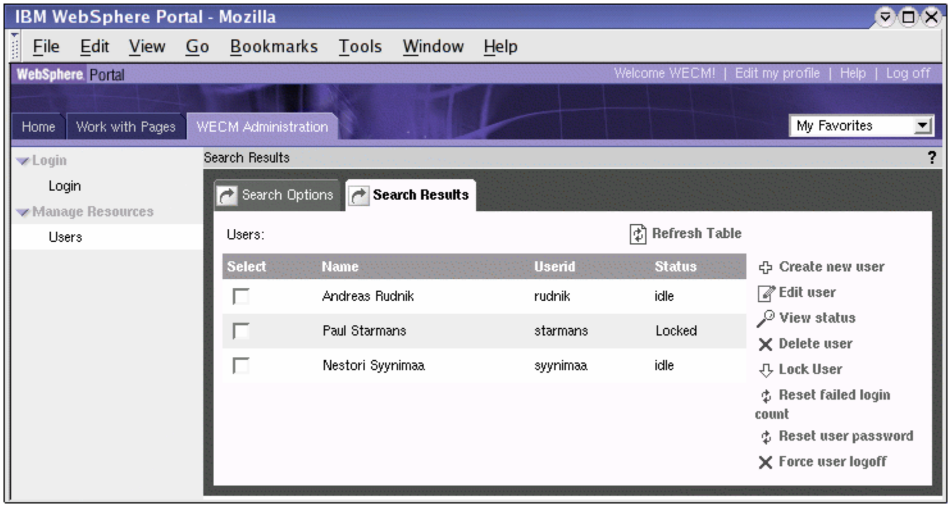
E-mail address:

Device ID: 0x

Figure 8-38 Search options

Click **Reset search** to set all fields to their default values. Click **Search** to start the search. If there are no results matching the search criteria, the portlet will return to the Search Options page. Otherwise, the portlet will display the a list of results in the Search Results page.

In the Search Results page, you can create new users, modify the attributes of an existing user, delete users, reset the failed login count and the password of a user, and force a user to log off. All these actions can be performed for a single user or for a group of users.



Select	Name	Userid	Status
□	Andreas Rudnik	rudnik	idle
□	Paul Starmans	starmans	Locked
□	Nestori Syynimaa	syynimaa	idle

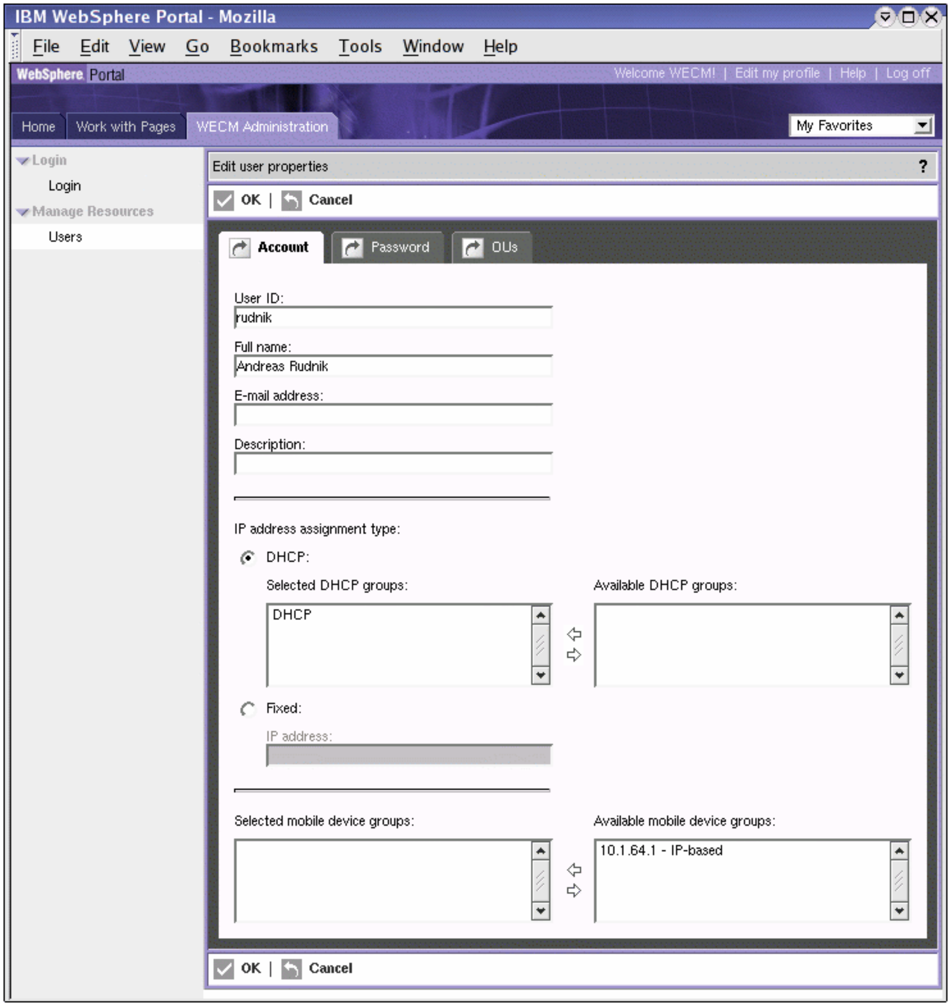
Figure 8-39 Search result

Create new user

Select **Create new use**r to add a user to the Connection Manager. Provide the user ID and full name. All other fields are optional.

IP address assignment type specifies whether an IP address is dynamically allocated from a Dynamic Host Configuration Protocol (DHCP) group when the user connects to the Connection Manager, or whether a fixed IP address is assigned to the user ID.

Selected mobile device groups specify that the user ID is restricted to connect with the Connection Manager using only the selected mobile device contained within a mobile device group.



User ID: rudnik

Full name: Andreas Rudnik

E-mail address:

Description:

IP address assignment type:

● DHCP:

Selected DHCP groups: DHCP

Available DHCP groups:

○ Fixed:

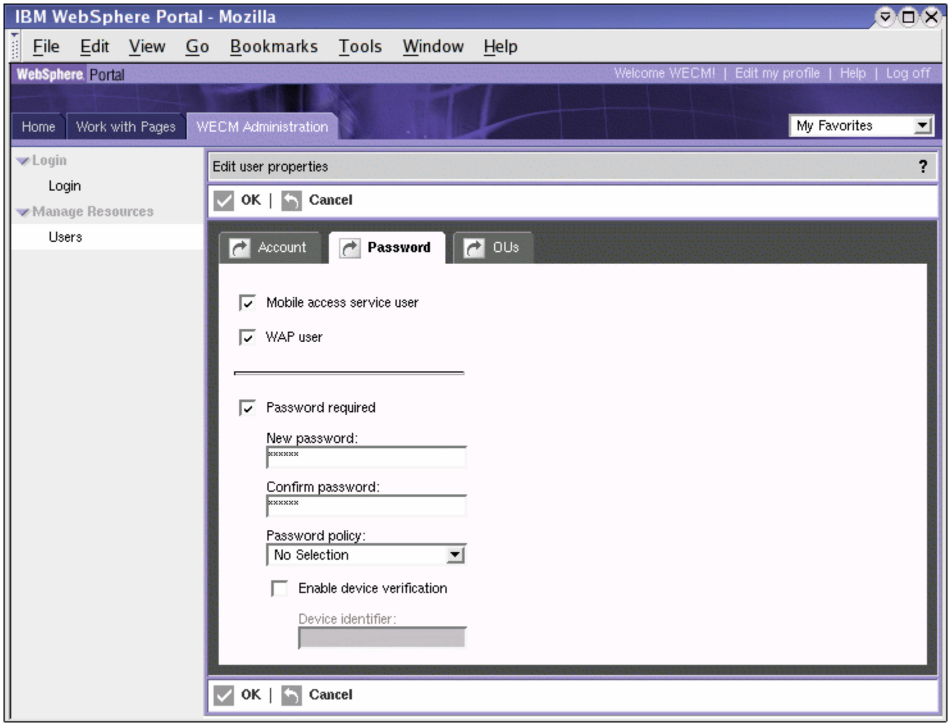
IP address:

Selected mobile device groups:

Available mobile device groups: 10.1.64.1 - IP-based

Figure 8-40 Create user - account details

Click the **Password** tab to set password related attributes. In this screen, you can specify whether the user can connect to the Connection Manager through Mobile access services and/or WAP. Use the password fields to set the initial password for the user and select the password policy.



Edit user properties ?

☑ Mobile access service user

☑ WAP user

☑ Password required

New password: ******

Confirm password: ******

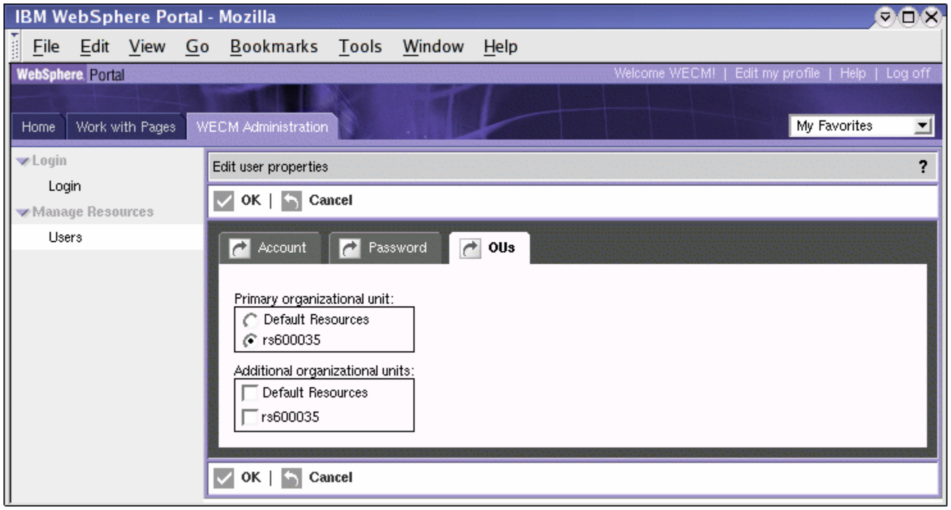
Password policy: No Selection

□ Enable device verification

Device identifier:

Figure 8-41 Create user - password settings

Select the **OUs** tab to set primary Organizational Units in which the user will be created. You can also select additional Organizational Units.



Primary organizational unit:
○ Default Resources
● rs600035

Additional organizational units:
□ Default Resources
□ rs600035

Figure 8-42 Create user - Organizational Unit

Click **OK** to create the user. The portlet will return to the Search Results screen.

Edit user

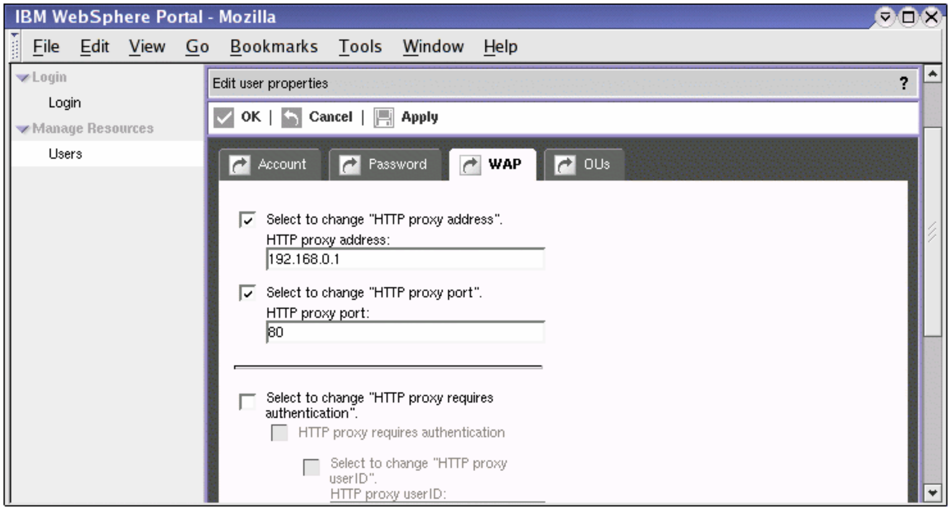
Select a user from the list and click the **Edit user** button. First, the account screen is presented. All the settings that can be specified during the creation of a user can be changed in the Edit user mode. Additionally, the following user attributes can be set.

The **Account** tab allows you to lock or unlock the user account and enable tracing for the user. Trace data for a user is sent to the trace file named in the Connection Manager Logging tab. The default file name is /var/adm/wg.trace.

The **Password** tab lets you set a password expiration date. The user account will then be locked by the specified data. You can also select to force a password change. If enabled, the user has to change the password when one logs in the next time.

The **WAP** tab allows you to override the default WAP proxy setting for the HTTP proxy.

If you choose to change the attributes for more then one user, then each attribute will have an additional check box. This check box must be selected before you can change the attribute.


IBM WebSphere Portal - Mozilla
File Edit View Go Bookmarks Tools Window Help
Login
Login
Manage Resources
Users
?
OK | Cancel | Apply
Account
Password
WAP
OUs

Edit user properties

☑ Select to change "HTTP proxy address".
HTTP proxy address:
192.168.0.1

☑ Select to change "HTTP proxy port".
HTTP proxy port:
80

□ Select to change "HTTP proxy requires authentication".
□ HTTP proxy requires authentication
□ Select to change "HTTP proxy userID".
HTTP proxy userID:

Figure 8-43 Edit multiple users

View status

Select one or more users and click **View Status** to see the following attributes:

- Failed connection attempts
- Date of last failed connection
- Date of last password change
- Account status

Delete user

Select one or more users and click **Delete user**. This will permanently delete the user accounts. Note that the portlet does not ask for any confirmation.

Lock user

Select one or more users and click **Lock user** to lock their user account. The users will not be able to connect to Connection Manager until the account is unlocked.

Reset failed login count

You can use this function to set the counter of failed login attempt to zero.

Reset user password

This function is used to reset the password of one or more users. You have to specify a new password for the users. The **Force password change** flag will be set.

Force user logoff

Use this function to terminate a user connection to the Connection Manager.

8.3 User management

This section provides information on the way user details are stored in a directory server, and how these details may be used for creating resources.

This section describes the following:

- Extracting user information from a directory server
- The format of stored user details
- How to create large numbers of resources

8.3.1 User storage

As with all Connection Manager V5 resources, users are stored in the LDAP directory structure through an appropriate LDAP browser such as DMT; it is possible to view and modify user attributes. However, be aware that modifying users in this way will not invoke the error checking that the Gatekeeper applies to resource changes.

8.3.2 Extracting user data

For purposes of portability, records stored in the directory's database may be exported to a Lightweight Directory Interchange Format (LDIF) file. The following example describes the method used for creating an LDIF file using IBM Directory Services V4.1. This process may also be applied to versions earlier than V4.1. For other directory servers, consult the documentation for details. Two different methods may be used to complete this task:

- From a command line using the command **`db2ldif`**
- Through the Web browser interface, if configured

Command line

To create an LDIF file from the command line interface, complete the following steps:

1. From a command prompt, locate the directory containing the **`db2ldif`** command:
   - UNIX: <LDAP_HOME>/sbin
   - Windows: <LDAP_HOME>/bin

2. Run the command **`db2ldif -o <output_file>`**. The result of the command is that the entire contents of the directory will be written out to the output file in LDIF format. If only the users are required, append **`-s <user_subtree_DN>`** to the end of the command.

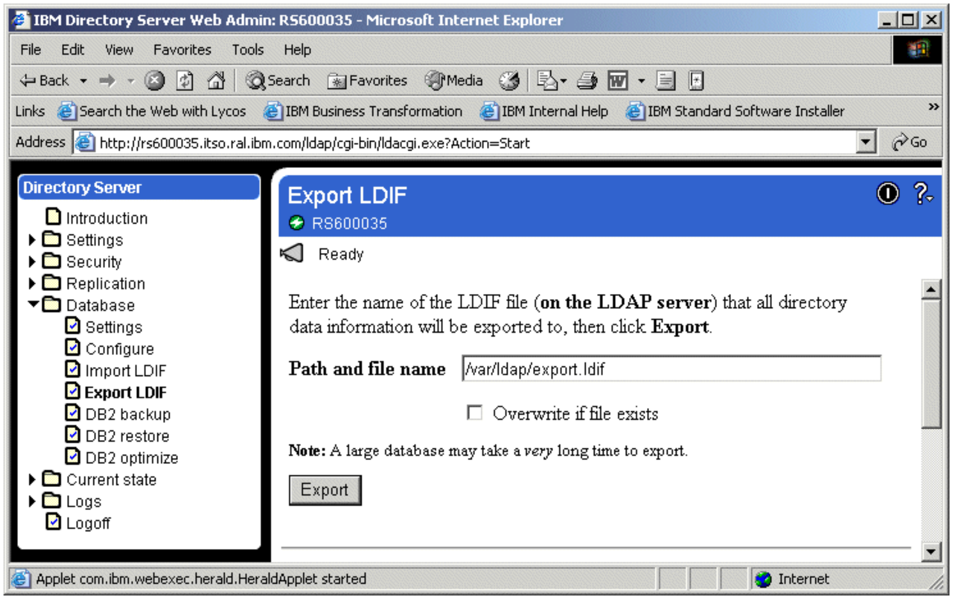
> **Note:** The option to specify a subtree is only provided in the command line tool, and not through the Web browser.

3. Using a text editor, browse the output file to ensure the command was successful.

Web browser

These are the steps:

1. Through a suitable Web browser, open the URL: `http://<LDAP_hostname>/ldap.`
2. Enter the required user ID and password to log on.
3. Click **Database -> Export LDIF**.
4. As shown in Figure 8-44, either accept the default export destination, or enter an alternate path. If an alternative is used, ensure the directory exists prior to continuing. If an existing LDIF file can be overwritten, check the relevant box to indicate this. Click **Export** when ready.


IBM Directory Server Web Admin: RS600035 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Search the Web with Lycos IBM Business Transformation IBM Internal Help IBM Standard Software Installer
Address
Go
Directory Server
Introduction
Settings
Security
Replication
Database
Settings
Configure
Import LDIF
Export LDIF
DB2 backup
DB2 restore
DB2 optimize
Current state
Logs
Logoff
Export LDIF
RS600035
Ready
Applet com.ibm.webexec.herald.HeraldApplet started
Internet

Address: http://rs600035.itso.ral.ibm.com/ldap/cgi-bin/ldacgi.exe?Action=Start

Enter the name of the LDIF file (**on the LDAP server**) that all directory data information will be exported to, then click **Export**.

Path and file name /var/ldap/export.ldif

□ Overwrite if file exists

Note: A large database may take a *very* long time to export.

Export

Figure 8-44 Export LDIF

5. Depending on the amount of data stored in the directory, this process may take some time to complete as it is not possible to limit the output to a particular subtree. Check the progress messages to ensure that there are no errors.
6. If this process needs to be repeated, click **Clear results** and repeat the steps.

8.3.3 Creating LDIF files

The files obtained in 8.3.2, “Extracting user data” on page 133 can be quite useful for creating new LDIF files containing a large number of users, which can later be imported into the directory. Prior to this, it is important to understand the format of the files, and how this can be used to create new users.

Connection Manager user format

Example 8-2 shows an example of a single user taken from the user directory. This was obtained from an LDIF file exported through the Web interface.

Example 8-2 User details

```
dn: uid=starmans,cn=users,ou=sun4,dc=itso,dc=com
ibm-wgclient: TRUE
ibm-deviceidverify: FALSE
authreq: 1
httpproxyport: 80
ipaddr: 0
hproxyauth: 0
ibm-wapclient: TRUE
trace: 0
locked: 0
addresstype: 1
ibm-tismstatus: C
addresspool: cn=DHCP Group,ou=sun4,dc=itso,dc=com
admchg: 1
cn: Paul Starmans
lastchg: 1059760546
objectclass: wlUser
objectclass: inetOrgPerson
objectclass: organizationalPerson
objectclass: person
objectclass: top
oldpasswords: {SHA}fX8Et+iVnBR3f7EwWloQMpp4tEg=
sn: 3
uid: starmans
userpassword: carla
failed: 1
lastfail: 1060285059
description: Wireless user
```

```
devicepool: ipaddr=9.24.104.180,ou=sun4,dc=itso,dc=com
mail: starmans@myemail.com
```

By comparing the details in Example 8-2 to the view of the same user through the Gatekeeper, it is easy to see how many of the fields match.

Using an LDIF to create Connection Manager resources

Once the format has been identified, creating your own LDIF file is an effective way of adding large numbers of resources to Connection Manager.

Using Example 8-2 as an example, by changing the `uid` in the first line from `uid=starmans` to `uid=user1`, and then importing this back into the LDAP, this now creates a new user called `user1`, while retaining all the same values as defined for the original user. However, any of the values may be changed as appropriate, but you must ensure that the user ID is unique for each user.

By extending this example, you may create many users in the same LDIF file. The basic steps are:

1. Export the directory to an LDIF file using the steps described in 8.3.3, “Creating LDIF files” on page 135. Keep an original copy of this file, which may be used later should a clean restore of the directory be required.
2. Obtain a single users details in LDIF format. Caution should be taken to ensure that only items belonging to the required user are selected.
3. Open the LDIF file using a text editor.
4. Using the single user as a guide, create additional users by altering the `uid` and any other relevant fields, such as common name, mail, and password.
5. Import the new LDIF file into the LDAP either by running the **`ldif2db`** command for through the Web interface. Depending on the number of resources being created, this may take some time to complete.

As with any LDIF import of Connection Manager resources, the Gatekeeper does not perform error checking during the import as it is done outside the control of Connection Manager. While any LDAP errors will be shown such as schema violations, Connection Manager errors will not.

> **Note:** It is a good idea to schedule regular backups of the LDAP directory. This will help to reduce the risk of losing large amounts of configuration data in the event of a failure.

9

Mobile Access Services

This chapter describes Mobile Access Services of WebSphere Everyplace Connection Manager. In this chapter we introduce the concept of Mobile Access Services, and which resources are available for Mobile Access Services.

This chapter also shows how to add Mobile Access Services and other resources to enable Mobility Clients to connect to Connection Manager.

In this chapter, the following topics are discussed:

- Mobile Access Communication
- Mobile Network Interface (MNI)
- Mobile Network Connection (MNC)
- Other resources for Mobile Access Services

9.1 Mobile Access Services communication

Mobile Access Services integrate data access from wireless and wireline networks so that applications and data can be made available to a mobile workforce. The Mobile Access Services support a wide range of wireless and dial-up networks such as wireless LAN and dial-up connection.

A basic Mobile Access Services configuration is illustrated in Figure 9-1. The mobile device (such as an IBM ThinkPad®) with the Mobility Client software is on the left and uses a wireless, dial-up, or LAN connection. The mobile device connects through the network to the Connection Manager and other servers, which give the mobile device access to enterprise applications and the Internet. In effect, the connection between the Mobile Access Services and Mobility Clients is a proprietary virtual private network (VPN).

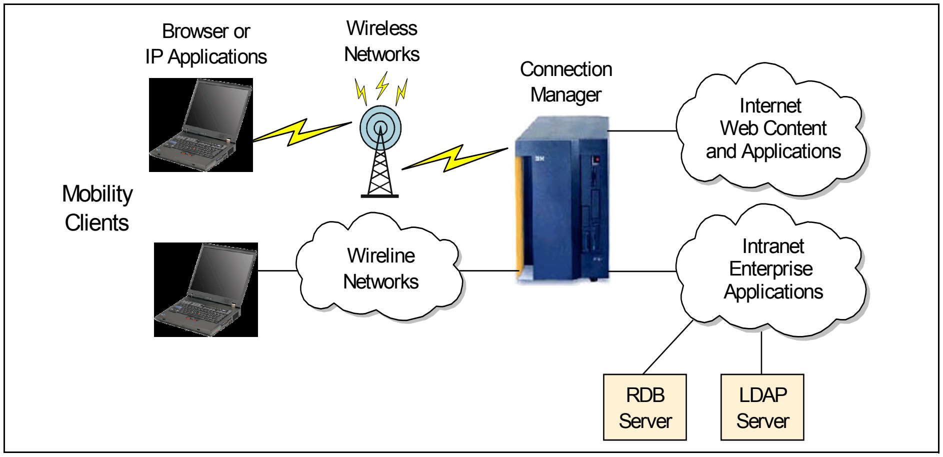

Browser or IP Applications
Wireless Networks
Connection Manager
Internet Web Content and Applications
Mobility Clients
Wireline Networks
Intranet Enterprise Applications
RDB Server
LDAP Server

Figure 9-1 Wireless and wireline communication

Existing application programs using a TCP/IP interface may use either wireless networks or wireline networks. Using TCP/IP integrates communication under common interface layer that shields network-specific details from the user application. Mobile Access Services through the connection to the Mobility Client provide network-specific enhancements such as:

- Data compression
- Data encryption
- Data optimization
- Authentication

Wireless applications using a non-IP packet-oriented connection such as Mobitex or DataTAC are supported by Mobile Access Services.

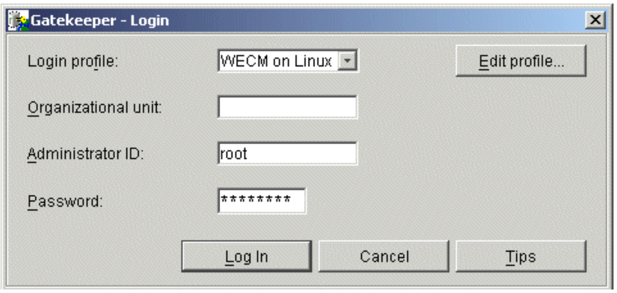
> **Note:** In this chapter, we will not discuss Mobility Clients. For more information about Mobility Clients, refer to Chapter 10, “Mobility Clients” on page 171.

9.2 Adding Mobile Access Services

To enable Mobile Access Services, you need to add Mobile Access Services to your Connection Manager.

To add Mobile Access Services, perform the following steps.

1. Start Gatekeeper with `wgcfg` on UNIX or select **IBM Gateway from Start -> Programs**.
2. Log in to Access Manager with your login profile such as shown in Figure 9-2.



Login profile: WECM on Linux

Organizational unit:

Administrator ID: root

Password: ********

Figure 9-2 Login to Access Manager through Gatekeeper

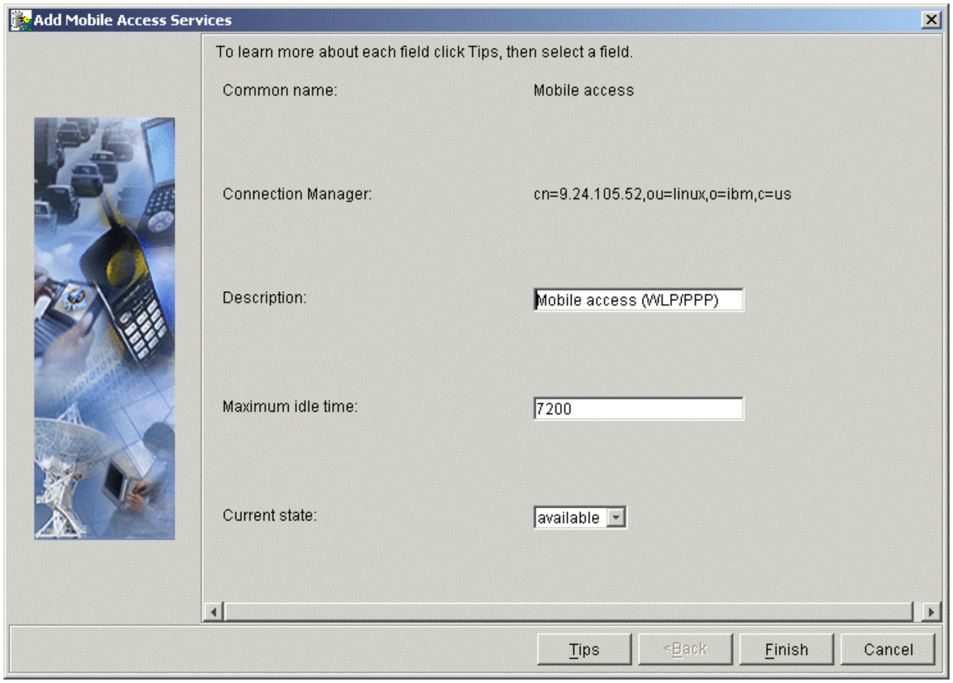
3. Right-click your **Connection Manager** in the left pane of the **Resources** tab and select **Add -> Mobile access.**
4. You will specify following properties:
   – Maximum idle time: 7200

     Specifies the maximum time of inactivity before the Connection Manager shuts down a Mobility Client. The default value is 7200 seconds.

   – Current state: Available

     Indicate current state of Mobile Access Services. The default is available.

   Leave them as default values and click **Finish**. You can modify these values through Gatekeeper. For detailed information refer to “Editing resource properties” on page 88.


Add Mobile Access Services

To learn more about each field click Tips, then select a field.

Common name: Mobile access

Connection Manager: cn=9.24.105.52,ou=linux,o=ibm,c=us

Description: Mobile access (WLP/PPP)

Maximum idle time: 7200

Current state: available

Figure 9-3 Adding mobile access services

9.3 Mobile Network Interface (MNI)

MNI is a resource assigned to a Mobile Access Services and defines an IP subnet, which is a contiguous range of IP addresses or groups of IP addresses, to support the number of Mobility Clients and mobile devices that can concurrently connect to the Mobile Access Services.

The same way TCP/IP functionality is presented through an IP interface, Mobile Access Services functionality is presented through a Mobile Network Interface (MNI). So, you activate the Mobile Access Services by creating a Mobile Network Interface. Similar to an IP interface, a Mobile Network Interface has a name, for example mn0 and an IP address, belonging to a mobile subnet, in which all mobile clients normally reside.

The Mobile Access Services route IP traffic for Mobility Clients and for WAP clients that use a native-PPP connections through IP subnet defined by MNI. Every Mobility Client or mobile device is assigned an IP address within the subnet of an MNI. These devices connect to your organization’s wired LAN through the MNI.

Mobile Access Services can have one MNI for all networks or multiple MNIs for different ranges of addresses. MNIs can support static addressing and Dynamic Host Configuration Protocol (DHCP) for a pool of dynamically assigned addresses.

Messaging clients and WAP clients that do not use native PPP to connect through a dial MNC and HTTP clients do not use MNI resources.

9.3.1 IP addressing concepts on MNI

When you define an MNI, you provide an IP address and a mask to define the subnet of the IP addresses available to Mobility Clients that will connect to Connection Manager through it. The subnet mask is applied to the address to specify a range of addresses defining a subnetwork within your network. You cannot extend the subnetwork beyond the range of your network.

For example, if you decide the following configuration for your MNI:

- IP address: 10.10.0.1
- Subnet mask: 255.255.255.0

The Mobility Clients and mobile devices that connect to Connection Manager will be given IP addresses between 10.10.0.2 and 10.10.0.254. The address 10.10.0.1 is reserved for the MNI. And in some network provider, the first and the last address from the entire ranges are reserved as network broadcast addresses as well. So, generally you can receive 253 different addresses for your Mobility Clients, which can connect to Connection Manager concurrently.

The subnet mask is similar in form to an address consisting of four octets, with all bits on except those at the end of the mask that indicate how many addresses are available. For example, a mask of 255.255.255.0 looks like:

```
11111111 11111111 11111111 00000000
```

while 255.255.240.0 looks like:

```
11111111 11111111 11110000 00000000
```

and allows for a larger range of host addresses.

In general, each time you change the last one to a zero, you double the number of addresses available. If you change more than one bit from a one to a zero, you might multiply the number of available addresses by two to the *n*th power, where *n* is the number of bits you changed. For example, changing from 255 (11111111) to 240 (11110000) multiplies the number of available addresses by two to the fourth power, or 16.

You can see some of examples of IP addressing with following Table 9-1. All these examples use a class B network, in which the first two octets define the network address, and the last two octets define the subnetwork and host addresses.

When you set MNI IP address to 34.34.130.1, here is the available range of IP addresses for Mobility Clients according to various the subnet masks after removing the individual addresses for the MNI and broadcast.

Table 9-1 Examples of IP addressing

Subnet mask	**Available range of IP addresses for Mobility Clients**
255.255.255.0 (11111111)	34.34.130.2 ~ 34.34.130.254 (253 addresses)
255.255.254.0 (11111110)	34.34.130.2 ~ 34.34.131.254 (509 addresses)
255.255.240.0 (11110000)	34.34.128.2 ~ 34.34.143.254 (4093 addresses)

After you acquire a subnet and define it as an MNI through the Gatekeeper, update your organization's routing tables to include the subnet being used for the MNI. You can either add static network route entries to individual machines in your network, or you can update a network router to include the MNI subnet addresses that should be routed through the Mobile Access Services.

For example, in Figure 9-4:

- Connection Manager IP address: 9.24.105.64
- Router IP address: 9.24.105.1
- MNI IP address: 10.10.0.1 with subnet mask 255.255.255.0

In order for traffic to be routed from the wired LAN to the MNI subnet, a routing table entry is made in which the subnet's destination IP address and mask (10.10.0.0 and 255.255.255.0, respectively) are routed through the Mobile Access Services (9.24.105.64).

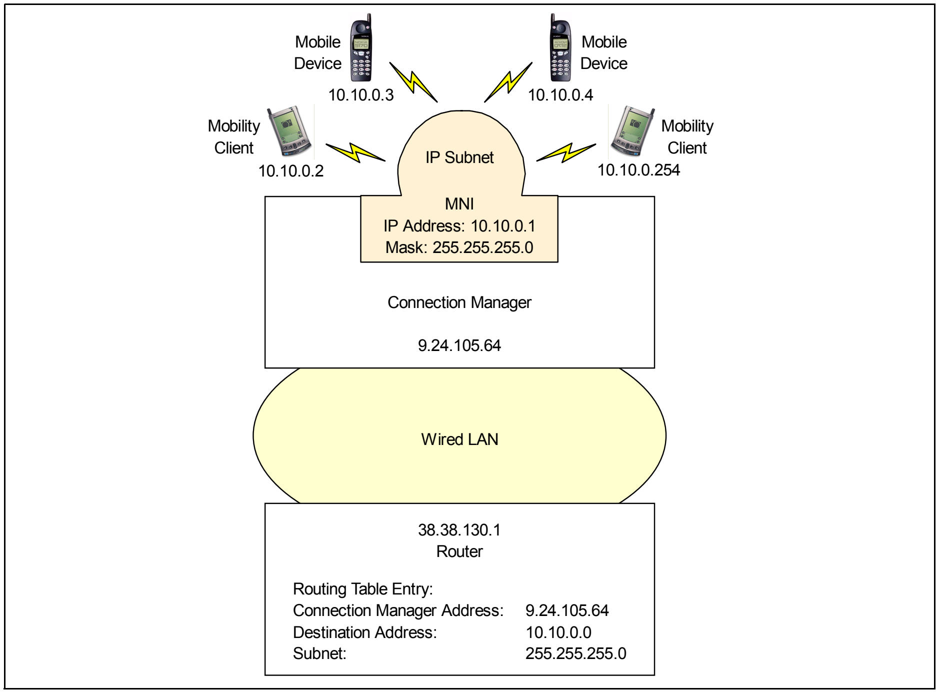
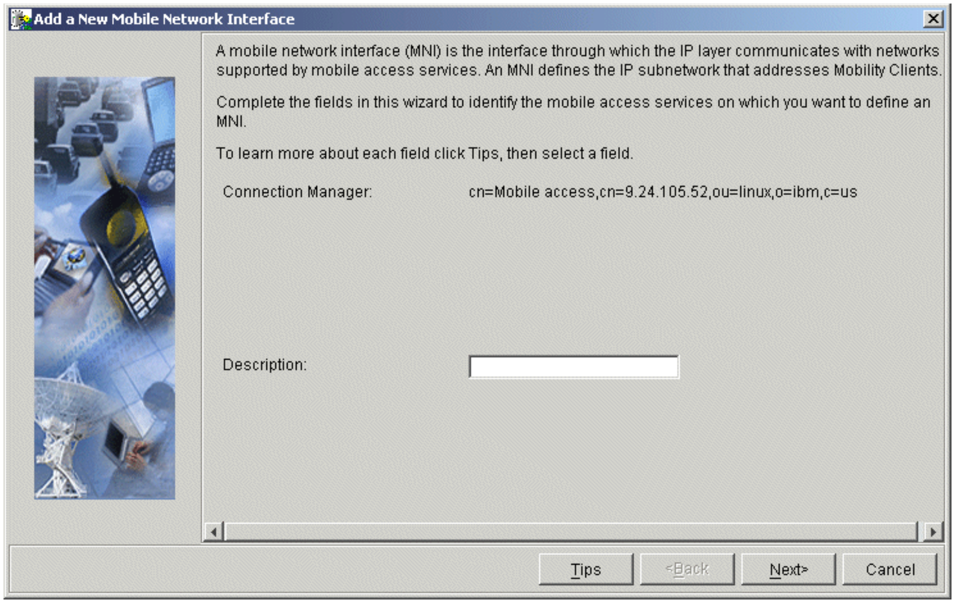


Figure 9-4 Example of MNI subnet and a routing table entry

9.3.2 Adding MNI to Mobile Access Services

To add MNI to your Mobile Access Services, perform following steps:

1. Start Gatekeeper and log in to Access Manager of your Connection Manager system. See Figure 9-2 on page 139.
2. Right-click on **Mobile Access Services** in the left pane of the Resources tab. Mobile Access Services is defined in Connection Manager. Select **Add -> Mobile network interface** from the Context menu. This brings up a wizard such as in Figure 9-5.



Add a New Mobile Network Interface

A mobile network interface (MNI) is the interface through which the IP layer communicates with networks supported by mobile access services. An MNI defines the IP subnetwork that addresses Mobility Clients.

Complete the fields in this wizard to identify the mobile access services on which you want to define an MNI.

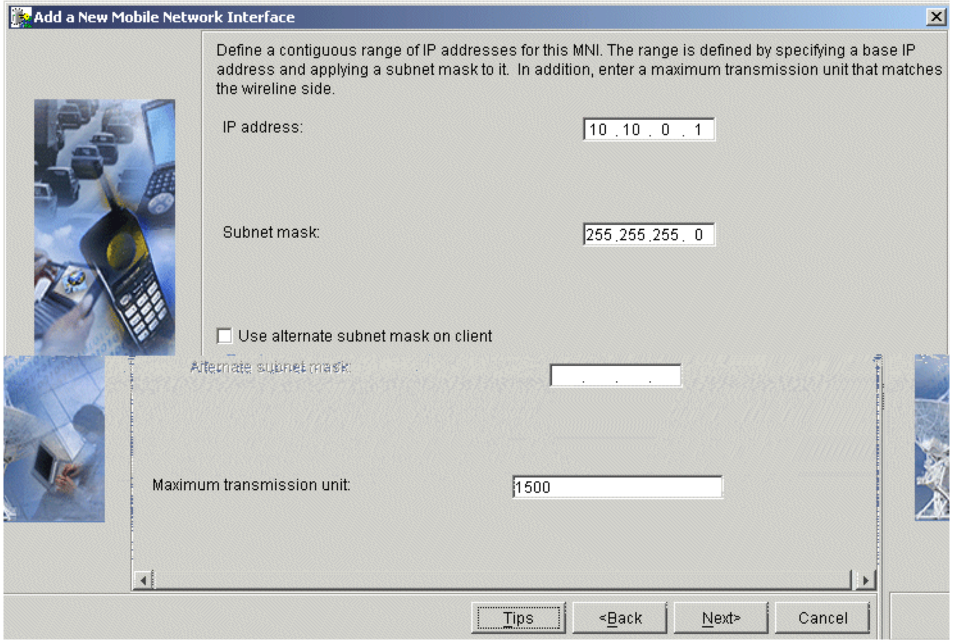
To learn more about each field click Tips, then select a field.

Connection Manager: cn=Mobile access,cn=9.24.105.52,ou=linux,o=ibm,c=us

Description:

Figure 9-5 Adding MNI to the mobile access services

Next, type the MNI IP address and subnet mask as you planned.



Add a New Mobile Network Interface

Define a contiguous range of IP addresses for this MNI. The range is defined by specifying a base IP address and applying a subnet mask to it. In addition, enter a maximum transmission unit that matches the wireline side.

IP address: 10 . 10 . 0 . 1

Subnet mask: 255 . 255 . 255 . 0

□ Use alternate subnet mask on client

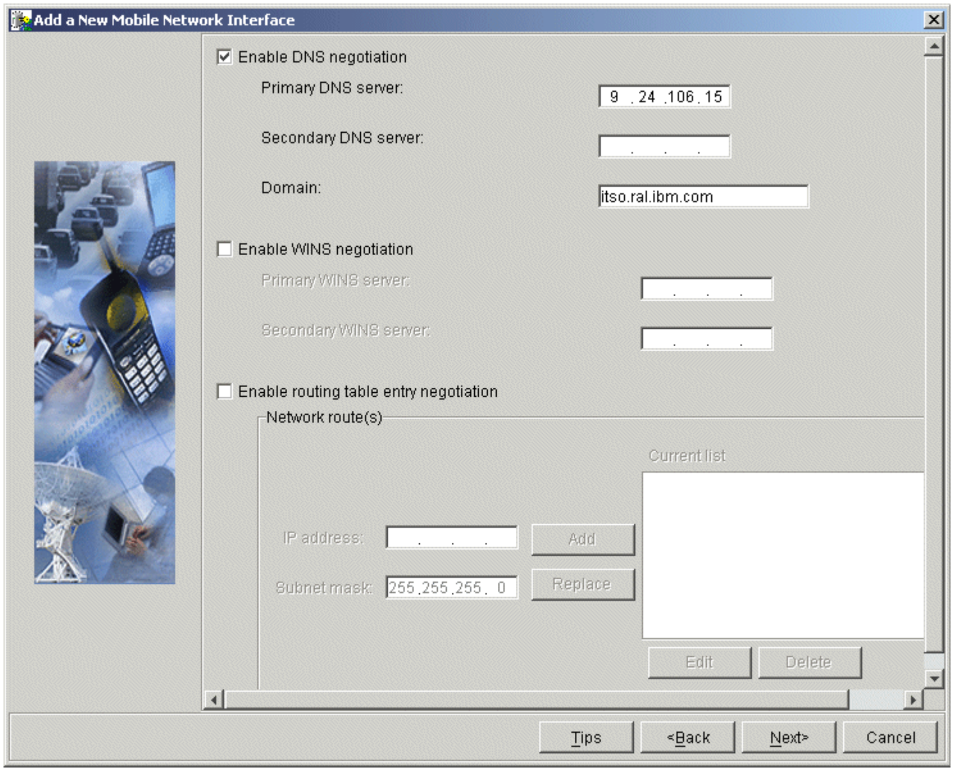
Alternate subnet mask: . . .

Maximum transmission unit: 1500

Figure 9-6 Mobile Network Interface - Step 2

   - MTU: Specifies the largest possible units of data (in bytes) that can be sent in a single frame.

3. In the next step, specify DNS or WINS IP for the name resolution of the Mobility Clients. Type `DNS IP` or `WINS IP` and optionally specify `Enabling routing table entry negotiation` and click **Next**.
   - Enable DNS negotiation: Specify the DNS IP address which will be used for the name resolution of Mobility Clients connecting to this MNI.
   - Enable WINS negotiation: Specify the WINS IP address which will be used for name resolution of Mobility Clients connecting to this MNI.



Add a New Mobile Network Interface

☑ Enable DNS negotiation

Primary DNS server: 9 .24 .106.15

Secondary DNS server: . . .

Domain: itso.ral.ibm.com

☐ Enable WINS negotiation

Primary WINS server: . . .

Secondary WINS server: . . .

☐ Enable routing table entry negotiation

Network route(s)

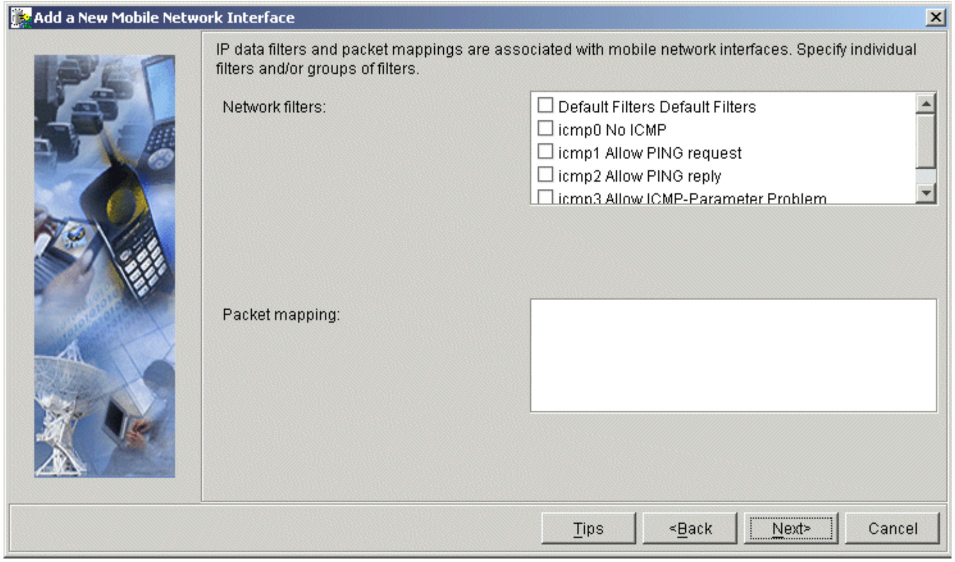
Current list

IP address: . . .

Subnet mask: 255.255.255. 0

Figure 9-7 Mobile Network Interface - Step 3

4. In this step, you can choose filters and packet mappings optionally. For your own filters and packet mappings configuration, you need to add a filter or packet mapping resource in advance. For detailed information refer to 9.5, “Other resources for Mobile Access Services” on page 154.


Add a New Mobile Network Interface

IP data filters and packet mappings are associated with mobile network interfaces. Specify individual filters and/or groups of filters.

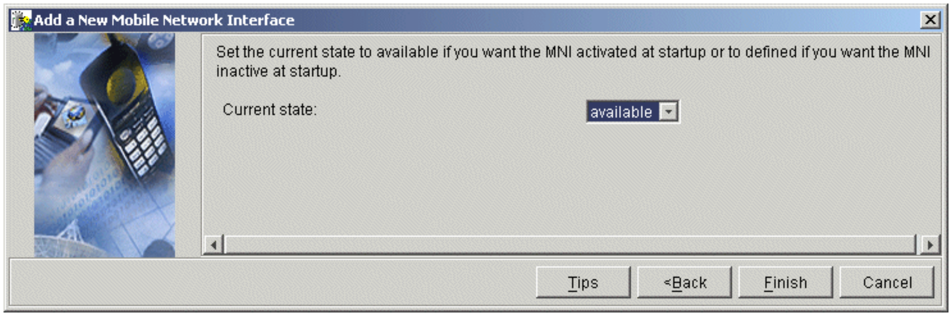
Network filters:

- □ Default Filters Default Filters
- □ icmp0 No ICMP
- □ icmp1 Allow PING request
- □ icmp2 Allow PING reply
- □ icmp3 Allow ICMP-Parameter Problem

Packet mapping:

Figure 9-8 Mobile Network Interface - Step 4

5. Select **Availabl**e from the drop down list. This makes your MNI active state and available. Click **Finish**.


Add a New Mobile Network Interface

Set the current state to available if you want the MNI activated at startup or to defined if you want the MNI inactive at startup.

Current state: available

Figure 9-9 Mobile Network Interface - Step 5

9.4 Mobile Network Connection (MNC)

A Mobile Network Connection (MNC) is a resources that is assigned to the Connection Manager and defines a specific type of network connection. The MNC consists of a line driver, a network protocol interpreter, and one or more physical ports. You configure one MNC for each network provider that you will use. You need MNCs for each network provider not only for Mobile Access Services, but also for WAP services and messaging services.

When you install the Connection Manager, you can install support for all networks, or for only the networks that you intend to use. MNC types include:

- Messaging connections, see Table 9-2 on page 147
- Mobile access connections, see Table 9-3 on page 148
- WAP connections, see Table 9-4 on page 149

Table 9-2 Available MKS for messaging services connections

MNC type	Description	Install this network support
ardis-tcp	Motient standard context routing (SCR) using TCP	Ardis
ardis-x25	Motient SCR using X.25	Ardis
ip-wdp	IP/UDP bearer adapter using wireless datagram protocol	Installed automatically with the Connection Manager
mobitex	Mobitex international standard connection using X.25	Mobitex
mobitex-tcp	Mobitex using TCP, usch as MObitex Internet application server(IAS)	Mobitex
rpa	RPA wireless messaging	SMS
sms-ois	Short message service (SMS) using open interface specification	SMS
sms-ois-x25	Short message service (SMS) using open interface specification over X.25	SMS
sms-smpp	Short message service to peer	SMS
sms-ucp	Short message service using universal computer protocol/external machine interface (UCP/EMI)	SMS

MNC type	Description	Install this network support
sms-smpp-x25	Short message service using SMPP Version 3.5 over X.25	SMS
sms-ucp-x25	Short message service using UCP/EMI over X.25	SMS
smtp	Simple mail transport protocol, as specified in RPC 821. Not that extensions to the RFC, such as MIME support or mail server authentication, are not supported	SMTP
snpp	Simple network transport protocol, as specified in RFC 1861. Note that all function of level one and required elements from level two are supported. SNPP level 3 is not supported.	SNTP
wctp		WCTP

Table 9-3 Available MNCs for Mobile Access Services using WLP and PPP

MNC type	Description	Install the network support
ardis-tcp	Motient standard context routing (SCR)	Ardis
ardis-x25	Motient SCR using X.25	Ardis
dataradio-bdlc	Dataradio base station data link controller (BDLC)	Dataradio
dataradio-msc	Dataradio multi-site controller (MSC)	Dataradio
datatac-5000	DataTAC 5000 using X.25	DataTAC
datatac-6000	DataTAC 6000 using TCP	DataTAC

MNC type	Description	Install the network support
dial-isdn	Integrated services digital network, including native-PPP connections (AIX only)	Dial
dial-ptsn	Public switched telephone network, including native-PPP connections	Dial
dial-tcp	Dial connection through IP-attached modem server, including native-PPP connections	Dial
ip-lan	IP-based network, such as CDPD, frame relay, connection with Internet service provider (ISP), or LAN	IP LAN
ip-wtls	Wireless Transport Layer Security connection	IP WTLS
mobitex	Mobitex international standard connection using X.25	Mobitex
mobitex-tcp	Mobitex using TCP, such as Mobitex Internet application server (IAS)	Mobitex
modacom-set	Modacom SCR using X.25	Modacom
rnc-3000	Radio network controller 3000	RNC-3000

* WLP: Wireless Link Protocol

* PPP: Point to Point Protocol

Table 9-4 Available MNCs for WAP proxy connections

MNC type	Description	Install this network support
ardis-tcp	Motient standard context routing (SCR) using TCP	Ardis

MNC type	Description	Install this network support
ardis-x25	Motient SCR using X.25	Ardis
dial-isdn	Integrated services digital network, only using native-PPP connections (AIX only)	Dial
dial-ptsn	Public switched telephone network, only using native-PPP connections	Dial
dial-tcp	Dial connection through an IP-attached modem server, only using native-PPP connections	Dial
ip-wdp	IP/UDP bearer adapter using wireless datagram protocol	Installed automatically with the Connection Manager
mobitex	Mobitex international standard connection using X.25	Mobitex
mobitex-tcp	Mobitex using TCP, such as Mobitex Internet application server (IAS)	Mobitex
sms-smpp	Short message service using short message peer to peer protocol (SMPP) Version 3.4	SMS
sms-ucp	Short message service using universal computer protocol/external machine interface (UCP/EMI)	SMS
sms-smpp-x25	Short message service using SMPP Version 3.5 over X.25	SMS
sms-ucp-x25	Short message service using UCP/EMI over X.25	SMS

MNC type	Description	Install this network support
smtp	Simple mail transport protocol, as specified in RFC 821. Note that extensions to the RFC, such as MIME support or mail server authentication are not supported	SMTP
snpp	Simple network paging protocol, as specified in RFC 1861. Note that all function of level one and required elements from level two are supported. SNPP level 3 is not supported	SNPP

Note: HTTP clients do not use MNC resources. For more detailed information, refer to Chapter 14, “HTTP Access Services” on page 275.

A Connection Manager that is configured exclusively as a subordinate node in a cluster does not need MNC installed unless it is a messaging MNC. For more information, refer to Chapter 22, “Clustering” on page 473.

9.4.1 Adding MNC for Mobile Access Services

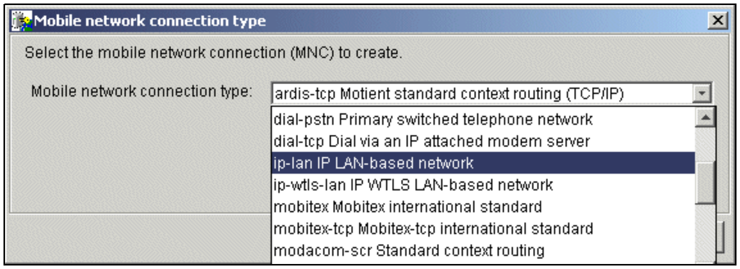
After adding the Mobile Network Interface, you need to add MNC to provide the interface between the Connection Manager and the wireless network. The MNC becomes a mean for communication between the Connection Manager and network provider of Mobility Clients and mobile devices.

There are number of different interfaces we support. Here we show you how to add an IP LAN-based network connection, which will allow you to establish a connection between a client that is connected to the LAN, and the Connection Manager that is connected to the LAN. For other supported networks, refer to Table 9-3 on page 148.

To add MNC for Mobile Access Services, perform following steps:

1. Start Gatekeeper and log in to Connection Manager through the login profile.

2. Right-click **Connection Manager** on the Resources tab and select **Add -> Mobile resources connection**. Refer to Figure 9-10 for details. At this point, select **IP-lan based network** from drop down list.



Mobile network connection type

Select the mobile network connection (MNC) to create.

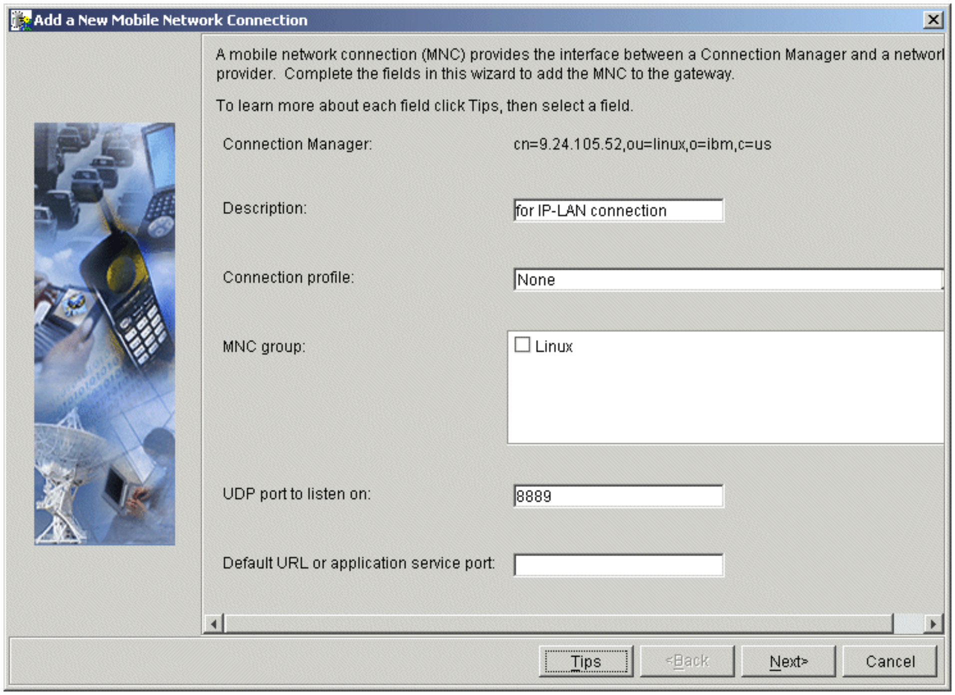
Mobile network connection type: ardis-tcp Motient standard context routing (TCP/IP)

dial-pstn Primary switched telephone network
dial-tcp Dial via an IP attached modem server
ip-lan IP LAN-based network
ip-wtls-lan IP WTLS LAN-based network
mobitex Mobitex international standard
mobitex-tcp Mobitex-tcp international standard
modacom-scr Standard context routing

Figure 9-10 Adding MNC - step 1

3. After selecting the Mobile Network Connection type, specify properties for this connection. You can specify the particular connection profile, which you have defined previously or provided by default. None is selected by default, and it means using the default connection profile.

   You can select **MNC group**, which assigns the user group who will use this MNC. Also, specify UDP port number. As a default it is 8889. For port numbers that are used for WebSphere Everyplace Connection Manager v5.0, refer to 2.4.2, “Use of ports” on page 25. See Figure 9-11.



Add a New Mobile Network Connection

A mobile network connection (MNC) provides the interface between a Connection Manager and a networl provider. Complete the fields in this wizard to add the MNC to the gateway.

To learn more about each field click Tips, then select a field.

Connection Manager: cn=9.24.105.52,ou=linux,o=ibm,c=us

Description: for IP-LAN connection

Connection profile: None

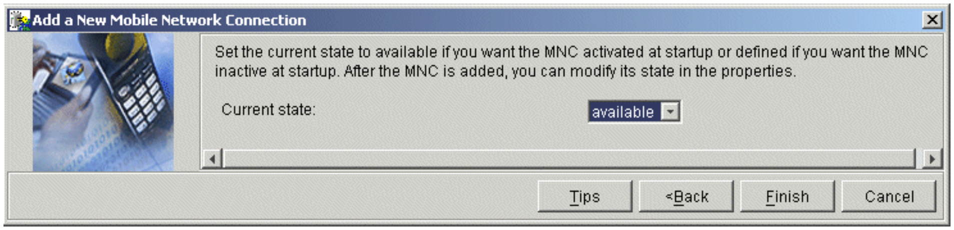
MNC group: □ Linux

UDP port to listen on: 8889

Default URL or application service port:

Figure 9-11 Adding MNC - step 2

4. Select **Available** to enable this MNC.



Add a New Mobile Network Connection

Set the current state to available if you want the MNC activated at startup or defined if you want the MNC inactive at startup. After the MNC is added, you can modify its state in the properties.

Current state: available

Figure 9-12 Adding MNC - step 3

You can add MNCs for other kinds of networks like messaging services and WAP services. For Messaging MNC, refer to Chapter 18, “Messaging services” on page 371. For WAP MNC, refer to Chapter 19, “WAP gateway” on page 415.

At this moment, you can connect to your Connection Manager from a Mobility Clients with a user ID. If you do not have a user ID, you need to add it for Mobility Clients. Refer to Chapter 8., “Administration” on page 85 for details on adding users to Connection Manager.

9.5 Other resources for Mobile Access Services

Resources that can be assigned to Mobile Access Services include:

- TCP-Lite
  - HTTP codec
- Connection profile
- Groups for mobile access services
- Mobile device
- Modem profile
- Network address translator
- Packet mapping
- Filter
- Routing alias

See Figure 9-13 for relations among above resources. It describes relations between each resources for wireless communication. For example, MNC applies a connection profile and a connection profile itself applies an authentication profile, TCP-Lite, and HTTP codec. You can select a connection profile or get one by default. You can apply none or several TCP-Lite services to a connection profile.

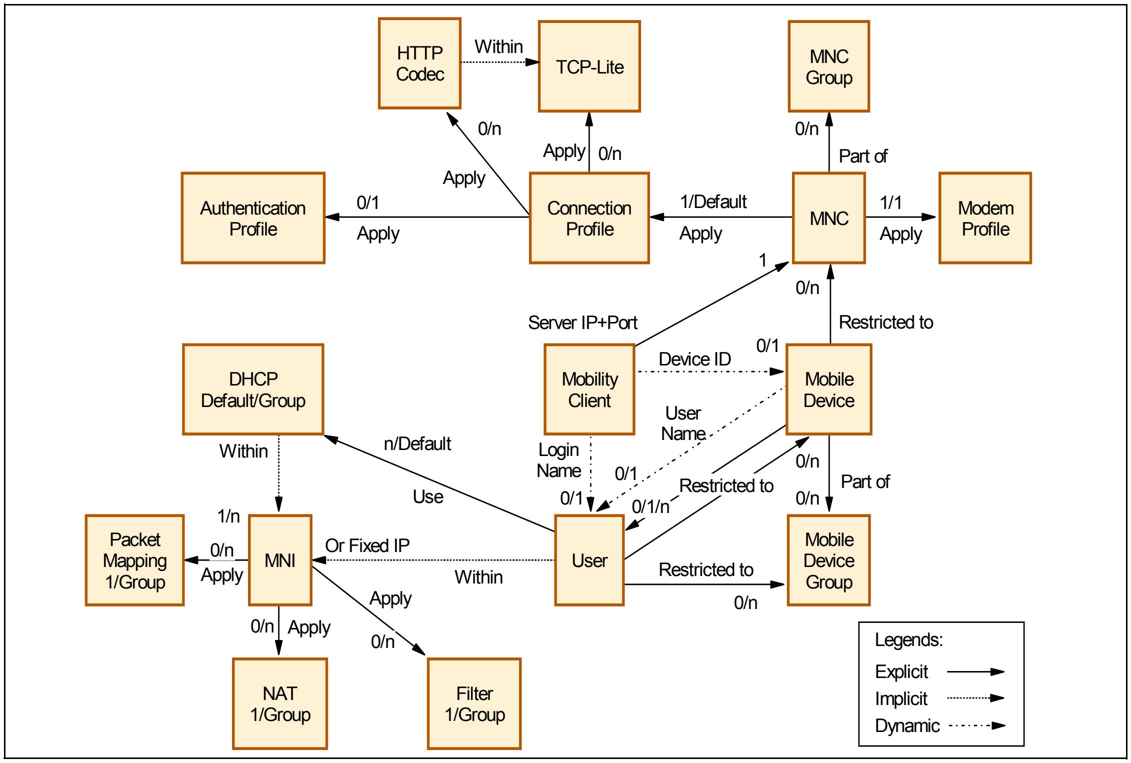


Figure 9-13 Relation of resources for mobile access services

9.5.1 TCP-Lite

TCP-Lite is a service that provides a transport channel that intercepts TCP in order to reduce the overhead involved in session management in which no application data is transmitted or received. TCP-Lite reduces or eliminates pure TCP protocol data units (PDUs) used in the setup, tear down, and acknowledgement of a channel while maintaining order, integrity, reliability, and security of the original TCP transport.

A TCP-Lite transport is applied to a connection profile, which is a set of configuration properties assigned to an MNC to control the performance options between an MNC and the Mobility Clients that connect to it.

You create the TCP-Lite object first through the Gatekeeper and set the connection profile to use TCP-Lite. After you apply that connection profile to your MNC, the TCP and IP header from TCP/IP packets between Connection Manager and Mobility Clients are to be removed and combined into WLP such as in Figure 9-14.

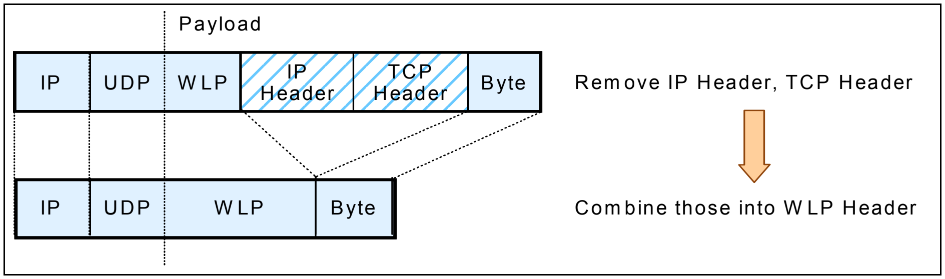


Figure 9-14 TCP-Lite packet change

HTTP codec

HTTP codec is a service that uses TCP-Lite as an underlying transport to provide a reduction in the over-the-air (OTA) byte count by removing and byte-encoding header fields in a HyperText Transport Protocol (HTTP) data stream.

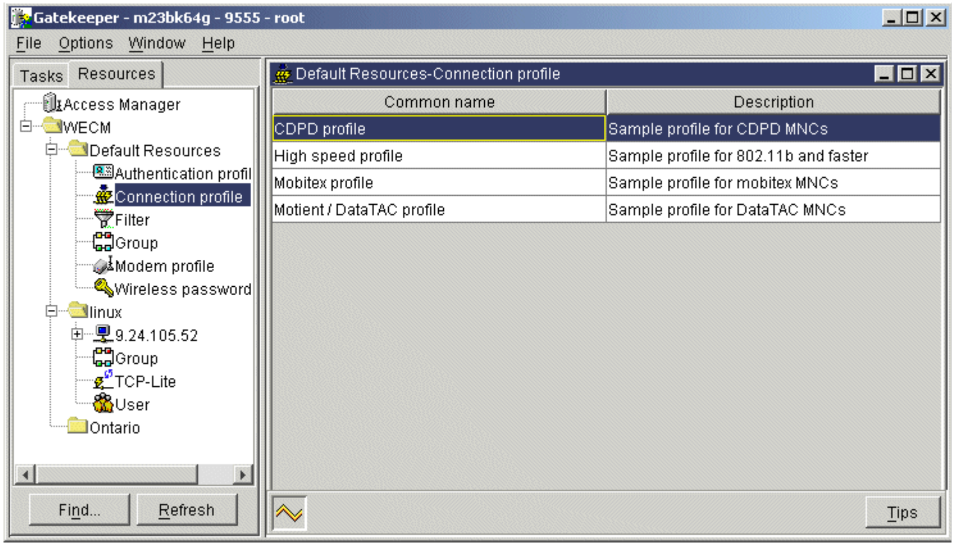
On the Mobility Client, an HTTP codec removes or encodes HTTP request headers, transmits the HTTP data stream, then reconstitutes the request headers at the Connection Manager before passing the traffic to target Web servers. Then, the Connection Manager removes or encodes the HTTP response headers before passing the traffic to the Mobility Client.

9.5.2 Connection profile

A connection profile is a set of configuration properties that are assigned to an MNC to control the properties of connection between the MNC and Mobility Clients that connect to it. When you create an MNC or edit its properties, you assign a connection profile to it. If you select **None** among connection profile drop-down lists, it means you will use the default connection profile.

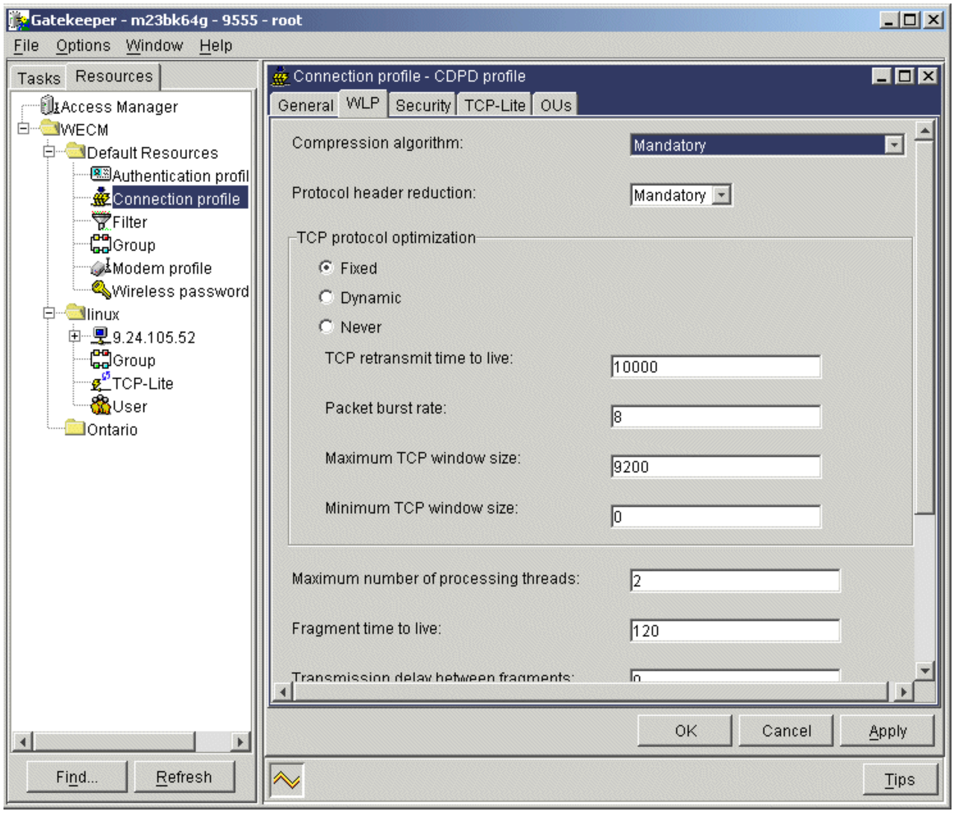
There are sample profiles provided with the Gatekeeper as shown in Figure 9-15. You can alter the sample profiles or create other profile, one for each set of properties that you want to assign to a given resource.

You can review the properties of connection profiles by double clicking each connection profile list on right pane of Gatekeeper.



Common name	Description
CDPD profile	Sample profile for CDPD MNCs
High speed profile	Sample profile for 802.11b and faster
Mobitex profile	Sample profile for mobitex MNCs
Motient / DataTAC profile	Sample profile for DataTAC MNCs

Figure 9-15 Sample connection profile



Compression algorithm: Mandatory

Protocol header reduction: Mandatory

TCP protocol optimization

● Fixed

○ Dynamic

○ Never

TCP retransmit time to live: 10000

Packet burst rate: 8

Maximum TCP window size: 9200

Minimum TCP window size: 0

Maximum number of processing threads: 2

Fragment time to live: 120

Transmission delay between fragments: 0

Figure 9-16 Sample CDPD connection profile - WLP tab

On WLP tab like Figure 9-16, you can specify the compression algorithm that the Mobility Clients are required to negotiate for this MNC:

- V42BIS

  A software implementation of the V.42bis compression standard algorithm. This implementation is provided for compatibility with Version 4 eNetwork Wireless Clients only.

- PKDCL

  The data compression libraries provided by PJWARE, Inc. This compression type can be used by all Mobility Clients using Windows operating systems with the exception of Windows CE.

- BSDZLIB

  The data compression libraries based on the Lempel-Ziv-Welch algorithm as provided by ZLIB. This compression type can be used by Mobility Clients using the Palm OS, Linux handheld, Linux desktop, and Windows CE operating systems.

- Optional

  No compression type is required for this MNC, and the Mobility Clients is free to negotiate any of the compression types.

- Mandatory

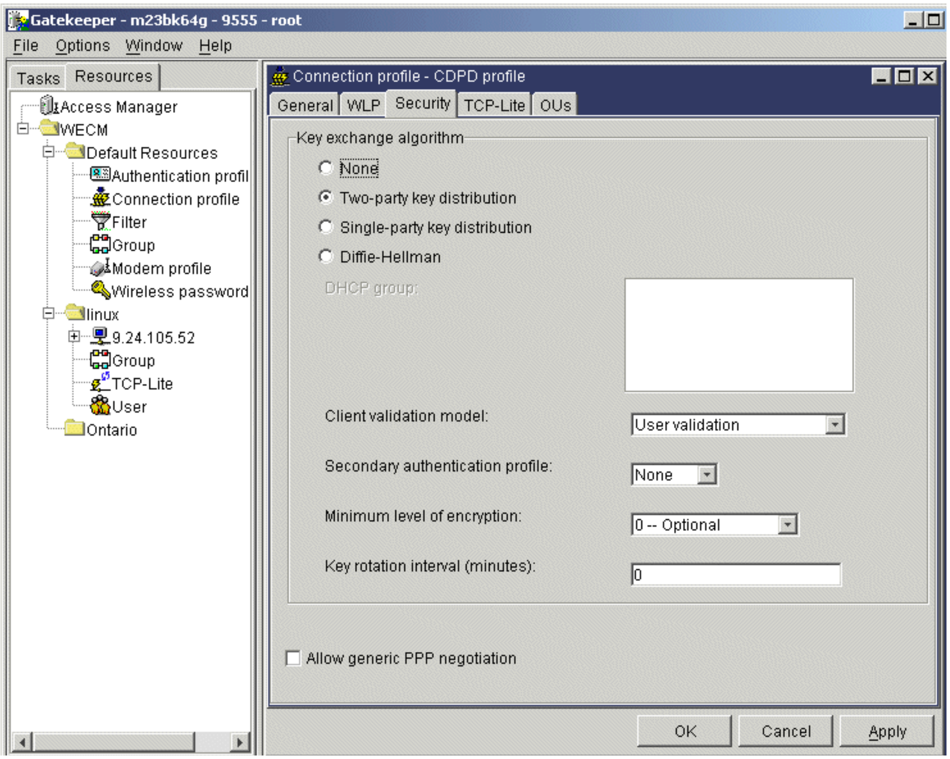
  Compression is required and any Mobility Client that does not negotiate a compression type is prevented from connecting to this MNC.

- Never

  Do not use any compression between the Connection Manager and the Mobility Clients. Over high speed network links, the time taken to compress and decompress data may actually create extra overhead, rather than saving on transaction times. This should be verified through testing prior to using this option.

Unless you know the operating system that Mobility Clients will use to log in to an MNC, do not pick a specific compression type. Choose either **Optional** or **Mandatory** to allow Mobility Clients on any operating system to negotiate the compression type. The default value is Optional.

On the **Security** tab as shown in Figure 9-17, you can specify the key exchange algorithm and client validation model of the Mobility Clients.



Key exchange algorithm

○ None
● Two-party key distribution
○ Single-party key distribution
○ Diffie-Hellman

DHCP group:

Client validation model: User validation

Secondary authentication profile: None

Minimum level of encryption: 0 -- Optional

Key rotation interval (minutes): 0

□ Allow generic PPP negotiation

Figure 9-17 Sample CDPD connection profile - security tab

Key exchange algorithm

The key exchange algorithm is used to validate Mobility Clients. Note that key exchange algorithm is assigned to an MNC and cannot be negotiated individually by Mobility Clients. All Mobility Clients connected through an MNC must use the same key exchange agreement.

- None

  The Connection Manager accepts a connection initiated by any Mobility Client using any device. When you define a connection profile with no validation, the Gatekeeper supplies a default user name (generic), which is used for accounting logging. The user ID for No validation is the user ID seen in the account file (wg.acct) for all Mobility Clients that connect through an MNC using this profile. This option is one which you do not need to define user IDs or mobile devices to the Gatekeeper.

  There are two MNCs that support only *No validation*:

  – Simple Network Paging Protocol (SNPP)

  - Simple Mail Transport Protocol (SMTP)

    > **Note:** In this case, the Mobility Client must be configured on the **Security** tab of the connection properties for the None Key Exchange.

- Two-party key distribution

  The Connection Manager is authenticated to the Mobility Client, and the Mobility Client is authenticated to the Connection Manager. You can specify an additional type of authentication by selecting a secondary authentication profile. If so, make sure that you define and choose an authentication profile other than the default system authentication profile.

  > **Note:** The Mobility Client must be configured on the **Security** tab of the Connections properties for the Password Key Exchange.

- Single-party key distribution

  The Mobility Client is authenticated to the Connection Manager. You can specify an additional type of authentication by selecting a secondary authentication profile. If so, make sure that you define and choose an authentication profile other than the default system authentication profile.

- Diffie-Hellman

  Both the Connection Manager and the Mobility Client are given the means to compute the same shared key. This option is one in which you do not need to define user IDs or mobile devices to the Gatekeeper.

  Using the Diffie-Hellman key exchange does not provide authentication. If you want to provide authentication, you must specify a secondary authentication profile.

  The Mobility Client must be configured on the Security tab of the Connection properties for Public Key Exchange.

For more detailed information about mobility security, refer to Chapter 12, “Mobility Client security” on page 211.

Client validation model

A client validation model which determines the validation level that is required when a Mobility Client initiates a connection with the Connection Manager. At connection time, the Connection Manager associates a user name with that client connection. This time is used for logging and can be identified in several ways: it can be entered at the Mobility Client as a user ID, derived from the identifier of the mobile device being used, or it can be a default value.

- User validation

  When a Mobility Client initiates a connection, the Connection Manager requires a user ID and an optional password. The user ID must be defined to the Gatekeeper or in a WebSphere Everyplace Server environment to IBM WebSphere Everyplace Subscription Manager. With a valid user ID, a person can initiate a session using any mobile device.

- User and device validation

  When a Mobility Client initiates a connection, the Connection Manager first validates the mobile device identifier, then requires a valid user ID, and an optional password. The Connection Manager checks whether this mobile device is associated with the supplied user ID. More than one mobile device can be associated with a user ID, and a device can be associated with more than one user ID. Use this model if you have devices assigned to more than one user, or if you have some devices assigned to one user and some shared.

- Device to user validation

  When a Mobility Client initiates a connection, the Connection Manager verifies the mobile device identifier, then derives the user ID that is assigned to that mobile device. Typically, this model is used for Mobility Clients that are configured not to display user IDs, and requires that each mobile device is assigned to just one user ID. A user, however, can be assigned more than one mobile device. This model requires that you define mobile devices and user IDs to the Gatekeeper.

Other than key exchange algorithm and client validation model properties, you can specify whether you will use TCP-Lite transport for this connection, and PPP clients such as generic PPP dialers are permitted to connect to the Connection Manager.

9.5.3 Groups for mobile access services

A group is a way to collect resources to use as a group rather than separately. You can create groups of the following types:

Broadcast

A list of recipients to which a broadcast message can be sent. A broadcast group can include users and MNCs. You can see how to add a broadcast group resource in 8.1.6, “Using broadcast groups” on page 112.

Mobile device

A pool of mobile devices that can be assigned to one or more users, eliminating the need to assign each device individually. A mobile device group is especially useful when those users share a pool of devices.

DHCP

A pool of IP addresses that can be assigned to users dynamically. You will assign a certain range of IP addresses to a DHCP group. When a user is created, a DHCP group may be assigned to the user to indicate the range of IP addresses that may be given to that user. A DHCP group can include IP addresses from different MNIs.

Filter

A list of filters to be applied together. If you have several MNIs, you can define a filter group, and then add filters to the filter group. You can apply the filter group to each MNI rather than assigning each filter to MNI separately.

Packet mapping or NAT

A list of packet mappings or network address translators (NAT) to be applied together. If you have several MNIs, you can define a group and then apply the group to each MNI, rather than assigning each packet mapping or network address translator separately.

9.5.4 Mobile device

A mobile device is device that clients use to communicate to the Connection Manager. You define mobile device to the Gatekeeper to control which devices can access your Connection Manager.

The information used to identify a mobile device depends on the network provider. You must have the device's unique identifier, which is often burned into the device or is firmware in the device.

WAP clients, such as WAP phones with microbrowsers, are not considered mobile devices unless they are connecting to the WAP proxy as generic-PPP accounts through a dial MNC.

9.5.5 Modem profile

A modem profile contains configuration information that enables the mobile access services to communicate with a public switched telephone network (PSTN) modem. The modem is attached to the Mobile Access Services and forms the gateway end of the link between the Mobile Access Services and the

Mobility Client. The modem profile contains default command and initialization strings for a particular modem. You specify modem profile when you create dial-pstn type of MNC.

The Mobile Access Services comes with several default modem profiles. You can modify the existing profiles or you can add a new profile.

If you use more than one modem profile, you must define a separate MNC for each profile.

9.5.6 Network Address Translator (NAT)

A Network Address Translator (NAT) is a resource that is assigned to an MNI. You use NAT to redirect traffic through a specified subnetwork represented by an MNI.

NAT lets the Connection Manager act as an agent between a public network and a private network. Based on RFC 1631, NAT lets you use IP addresses in a stub domain, which may be used in other stub domains. In a stub domain such as a corporate network that handles only origin or destination traffic from inside the network, there are very few IP addresses that need globally unique IP addresses. This aspect means that only a single, unique IP address is required to represent an entire group of computers.

The NAT defines a range of unique IP source addresses, then randomly assigns an originating packet to a port number (1024 through 65535). The NAT maintains the mapping of the packet to the port number in a translation table for the duration of a TCP session, or until a time out occurs for a TCP session or UDP connection.

External considerations

Your site routers must understand to route the NAT address to the physical NIC addresses of the gateway. Therefore, you will need to make sure that traffic destined for the NAT address can be routed to the machine where the Connection Manager is installed. You can accomplish this task by configuring a router, or by using the **arp** command to associate the NAT address with the MAC address of the machine. Run the **arp** command for each NAT address you want configured:

For AIX and Solaris the command looks like this:

```
arp -s ether <nat ip addr> <mac addr in colon sep format> pub
```

For a Linux system the command looks like this:

```
arp -v -i <ip interface> -s <nat ip addr> <mac addr in colon format> pub
```

For example:

```
arp -v -i eth0 -s 123.123.123.123 01:02:03:04:05:06 pub
```

where `eth0` is the IP interface name, `123.123.123.123` is the NAT IP address, and `01:02:03:04:05:06` is the MAC address of the machine.

Note: The **`netstat -i`** can be used to determine the MAC address of the physical adapters.

The **`arp`** commands will not survive a reboot of the machine. You can add the **`arp`** commands to initialization scripts so that they are executed after you restart the system. On AIX, the scripts are stored in /etc/rc.net or /etc/rc.wgated, and on Red Hat Linux distributions they are stored in /etc/rc.local.

There is no equivalent file on Solaris. However, you can perform a similar function. For example, you can create a script called /etc/init.d/ibmwgarp.sh with the commands in it, then create a symbolic link to it by issuing the command:

```
ln -s /etc/init.d/ibmwgarp.sh /etc/rc3.d/S99ibmwgarp.sh
```

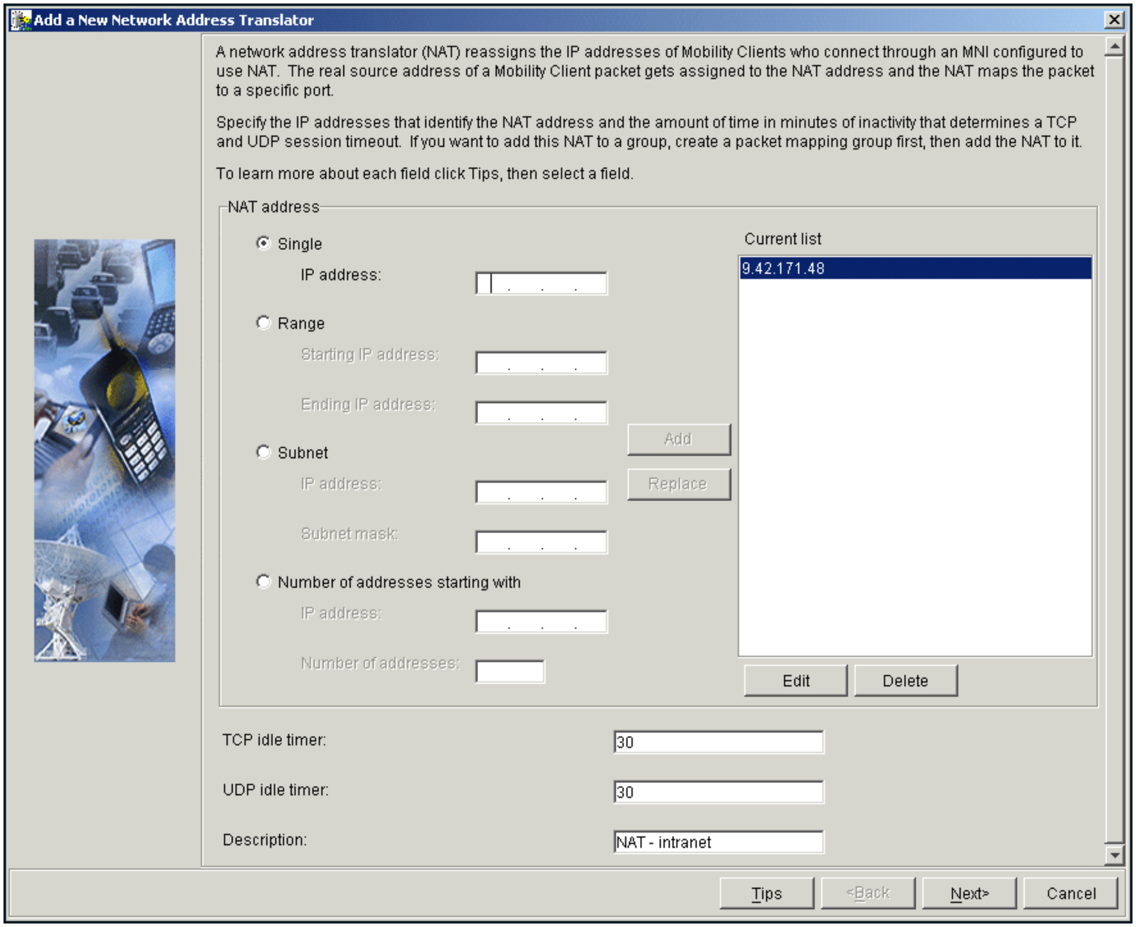
This will run the commands whenever the system is booted into runlevel 3 (the default).

NAT configuration

The IP address or addresses to add as the network address translation (NAT) source address is configured using the Gatekeeper as shown in Figure 9-18. The NAT source address is an IP address or a subnet of IP addresses that are routable to a physical network interface (and not addresses that are in the MNI subnet). The right column lists the current IP addresses defined to the NAT, which are identified in the left column as one or more ranges of dotted-decimal addresses:

- Single. A single address, such as 10.15.56.99
- Range. The lowest IP address through the highest IP address. For example, you can create a range of 40 addresses such as 10.15.56.10 10.15.56.50.
- Subnet. A subnet using an IP address and the mask. For example, you can define a 256 address subnet with a range of 34.34.73.0 through and including 34.34.73.255 by specifying 34.34.73.0 255.255.255.0.
- Number of addresses starting with the beginning address to include in a range followed by the number of addresses to include. For example, you can define a range of addresses from 34.34.73.0 through and including 34.34.74.3 by specifying 34.34.73.0 260.

Enter the IP address information, then click **Add** or **Replace** to list the IP addresses to include on the right. To modify addresses that are already in the list, click **Edit**. To erase an address from the list, click **Delete**.



Add a New Network Address Translator

A network address translator (NAT) reassigns the IP addresses of Mobility Clients who connect through an MNI configured to use NAT. The real source address of a Mobility Client packet gets assigned to the NAT address and the NAT maps the packet to a specific port.

Specify the IP addresses that identify the NAT address and the amount of time in minutes of inactivity that determines a TCP and UDP session timeout. If you want to add this NAT to a group, create a packet mapping group first, then add the NAT to it.

To learn more about each field click Tips, then select a field.

NAT address

● Single

IP address:

○ Range

Starting IP address:

Ending IP address:

○ Subnet

IP address:

Subnet mask:

○ Number of addresses starting with

IP address:

Number of addresses:

Current list

9.42.171.48

TCP idle timer: 30

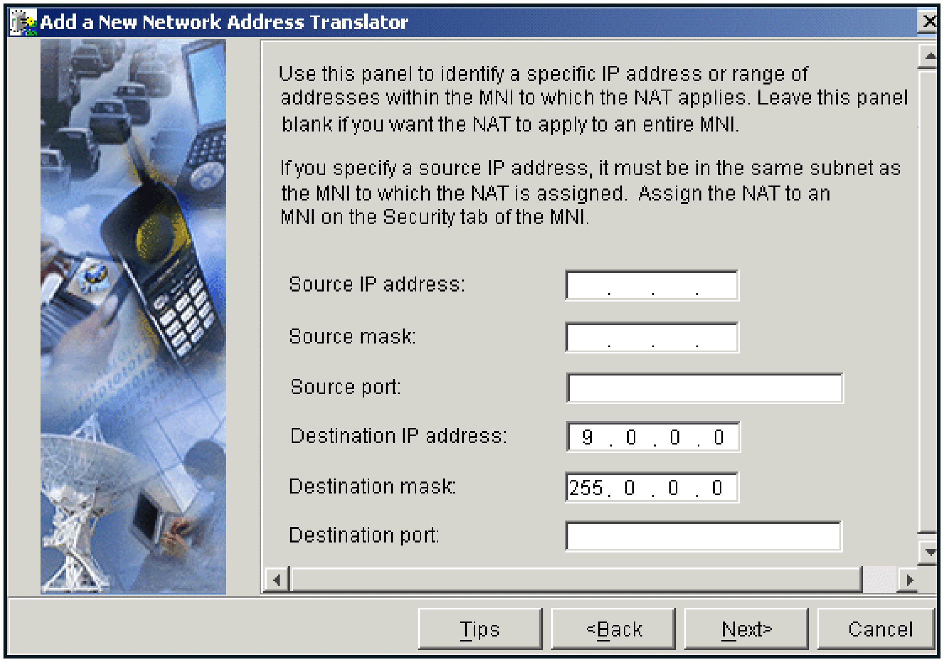
UDP idle timer: 30

Description: NAT - intranet

Figure 9-18 Network Address Translator (NAT) sample configuration

You also need to identify a specific address or range of addresses within the MNI to which the NAT definition applies.

Note: Leaving all fields blank means that all traffic will match the address translation filter as shown in Figure 9-19.



Add a New Network Address Translator

Use this panel to identify a specific IP address or range of addresses within the MNI to which the NAT applies. Leave this panel blank if you want the NAT to apply to an entire MNI.

If you specify a source IP address, it must be in the same subnet as the MNI to which the NAT is assigned. Assign the NAT to an MNI on the Security tab of the MNI.

Source IP address: . . .

Source mask: . . .

Source port:

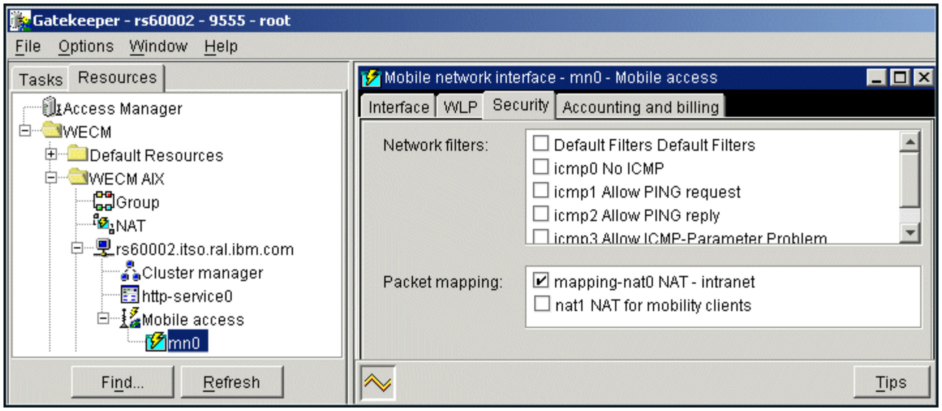
Destination IP address: 9 . 0 . 0 . 0

Destination mask: 255 . 0 . 0 . 0

Destination port:

Figure 9-19 Sample filter configuration

After creating the NAT resources, you need to update the MNI to associate the NAT definitions with the MNI as illustrated in Figure 9-20.



Gatekeeper - rs60002 - 9555 - root

Mobile network interface - mn0 - Mobile access

Network filters:
- □ Default Filters Default Filters
- □ icmp0 No ICMP
- □ icmp1 Allow PING request
- □ icmp2 Allow PING reply
- □ icmp3 Allow ICMP-Parameter Problem

Packet mapping:
- ☑ mapping-nat0 NAT - intranet
- □ nat1 NAT for mobility clients

Figure 9-20 Associating the NAT definitions to the MNI

When NAT is activated, the mobility access gateway will automatically add route entries for the NAT addresses to the MNI. This is required in order for the gateway to receive the mobile terminated traffic for processing back to its original settings and transmission to the Mobility Client device. In addition, Network Address Translator can be enabled or disabled through the MNI reference.

9.5.7 Packet mapping

Packet mapping is a resource that is assigned to an MNI. A packet mapping is a way to redirect some types of traffic through a specified subnetwork represented by an MNI. You can create packet mappings for four types of packets:

- TCP
- UDP
- ICMP
- Other

You can use packet mappings to modify some fields within the header of a packet. For example, you can set up a mapping to change the port number on outgoing TCP packets to a port on a mail server that has been optimized for Mobility Clients.

You can add packet mapping resource through the Gatekeeper. For further details, see 8.1.2, “Adding resources” on page 92.

Filter

A filter is a resource that is assigned to an MNI. Positive or negative filters are a way to control some types of traffic through a specified subnetwork represented by an MNI. For example, this particular packet flowing in this direction is allowed to flow (positive). Or, this particular packet flowing in this direction is not allowed to flow (negative).

You can create filters for four types of packets:

- TCP
- UDP
- ICMP
- Other

The filter criteria you use in a filter depends on the packet type. A filter can be defined to block packets or to pass packets according to the specified criteria. Filtering can be assigned to groups of users rather than the entire MNI. A set of default filters and a filter group called *Default Filters* are provided with Gatekeeper.

In addition to using filters to control data flow from the Mobile Access Services to the Mobility Client, you can also designate port numbers on the Mobility Client to prevent TCP or UDP traffic from flowing to the Mobile Access Services. To block outbound TCP or UDP traffic from the Mobility Client, edit the *artour.ini* file on Mobility Clients to add the keyword `TCPIP_Ports2Filter` or `UDPIP_Ports2Filter` followed by a space-delimited list of port numbers.

With added or default filters, you can specify a filter in the properties of your MNI resource through Gatekeeper.

Routing alias

A routing alias is a Mobility Client acting as a multihomed node to route data between the Mobile Access Services and a subnetwork which you specify. The Mobile Access Services delivers all traffic destined for the specified subnetwork to the Mobility Client's IP address. The Mobility Client then acts as a destination gateway, and routes the data to the destination address.

In the properties of MNI, you can specify the routing table entry for this purpose.

9.6 Enabling secure communication

The Connection Manager uses a modified Point-to-Point Protocol (PPP) called *wireless optimized link protocol* (WLP) to authenticate the connection between itself and Mobility Clients through a Mobile Network Connection (MNC). Each WLP MNC can use single-party key distribution, two-party key distribution protocol, or *Diffie-Hellman* to exchange keys and validate or authenticate Mobility Clients.

To view or change the type of key agreement used by the MNC between the Connection Manager and Mobility Client, edit the properties of the Connection profile assigned to the MNC, click the **Security** tab, then click the **Key exchange** algorithm field.

Client authentication certificates can be installed for WTLS support (Linux or Solaris systems only) and certificate-based authentication profiles. Determine if Mobility Clients should have a public key certificate installed on their systems.

For more detailed information about authentication and security, refer to Chapter 12, "Mobility Client security" on page 211.

10

Mobility Clients

This chapter provides information pertaining to the Mobility Client. The Mobility Client runs locally on the device, establishes an optimized mobile VPN, and enables cross-network roaming. Once Mobility Client authenticates to Connection Manager, a VPN is established and the device securely joins the enterprise intranet.

10.1 Overview

The Mobility Client is positioned below the TCP/IP stack and allows you to run IP applications across all supported networks. To the end user, a radio network becomes just another network that does not require any specialized communication protocol or programming interfaces.

Mobile application programs using TCP/IP interface have access to both wireless and wireline networks. Programmers can develop applications in a local area network (LAN) environment using the standard TCP/IP application programmer's interface (API), then can run the applications to the Connection Manager environment without modification.

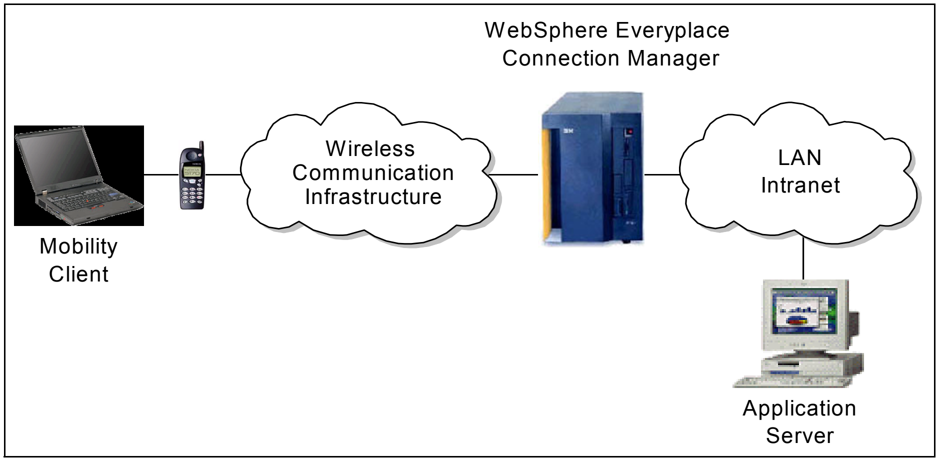


Figure 10-1 Mobility Clients

10.2 Everyplace Mobility interfaces

Everyplace Mobility software operates at the level of the network device drivers. It resides below the IP stack, and looks like a network adapter driver. Applications that uses the IP stack are not aware that it is operating over a mobility network.

10.3 Supported platforms

Mobility Clients are supported by the following platforms:

- Windows 95, 98 (SE Recommended), ME, NT Version 4.0 (Service Pack 4 or later), Windows 2000 and XP.
- Windows CE 3.0
  - Versions
    - H/PC 2.00
    - H/PC Pro 2.11 (recommended)
  - Processors
    - StrongARM
    - MIPS
    - SH3
- Pocket PC 2002/3
- Palm OS 4.x
- Linux Zaurus

10.4 Supported networks

Mobility Clients are supported in the following networks:

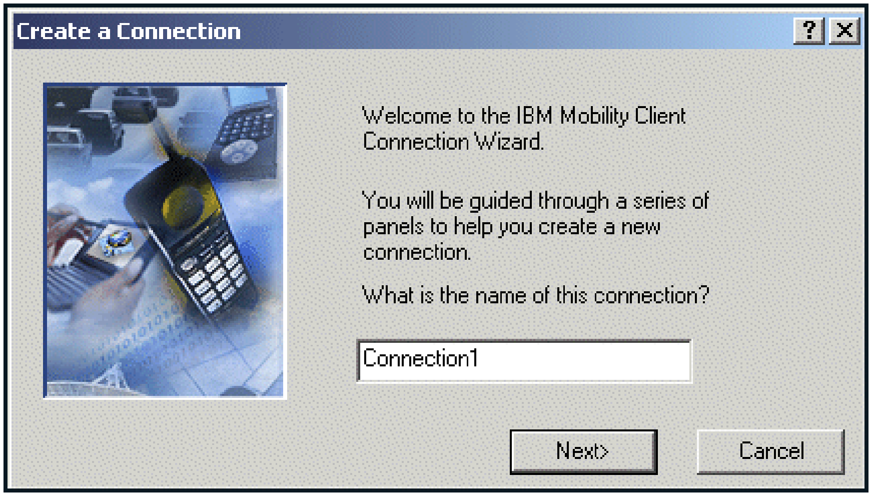
- IP-based, including CDPD, CDMA, and GPRS
- Dataradio
- DataTAC and Private Mobile Radio
- Mobitex
- Norcom Satellite
- PSTN Dial

10.5 Native Windows modem support

The Connection Manager architecture of the client takes advantage of native windows modem support (known as TAPI - Telephony API). The client uses the system services to interact with a network adapter (or modem).

10.6 Creating a connection

The Create a connection screen is the first to be displayed. In the Mobility Client connection wizard, provide the name of your connection.


Create a Connection

Welcome to the IBM Mobility Client Connection Wizard.

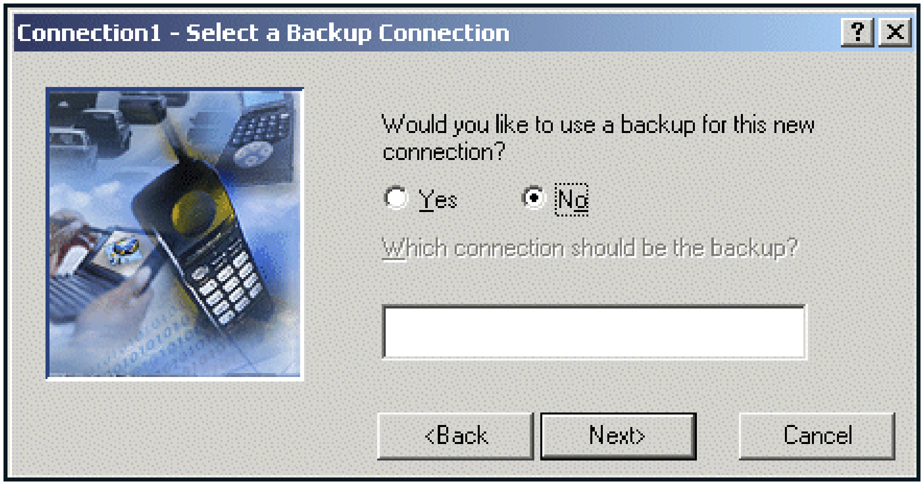
You will be guided through a series of panels to help you create a new connection.

What is the name of this connection?

Connection1

Figure 10-2 Create a connection

Once you have provided a connection name, continue by clicking the **Next>** button. The next screen displayed provides a way of selecting a backup connection. If you have other Connection Manager connections, they will be displayed and you can select the connection and select **Yes**. If not, the **No default** is selected, and you can continue by selecting the **Next>** button.


Connection1 - Select a Backup Connection

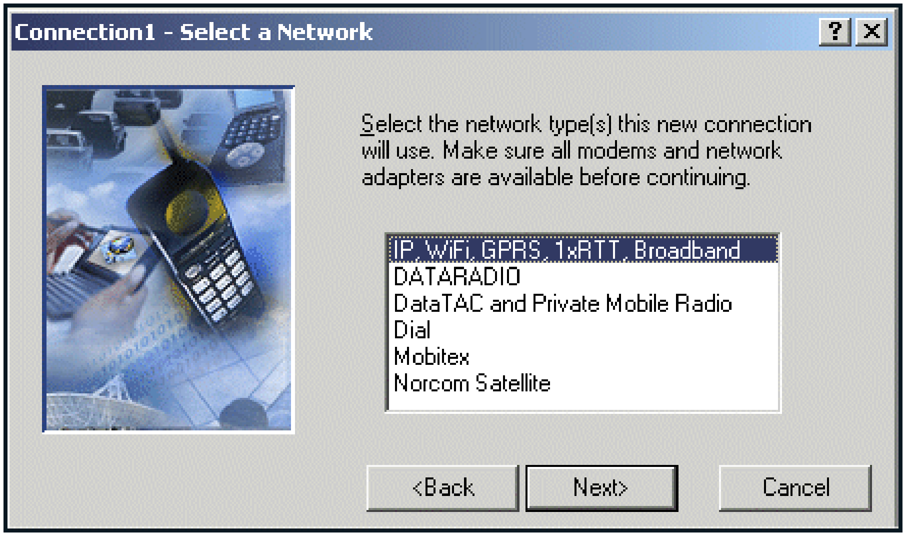
Would you like to use a backup for this new connection?

○ Yes ● No

Which connection should be the backup?

Figure 10-3 Backup connection selection

The next panel provided is the Select a Network panel. This panel provides the user the capability of selecting a network or networks that the device will use to connect to Connection Manager. Only networks that have been installed will be displayed. From this point on the configuration, panels are unique per network.


Connection1 - Select a Network

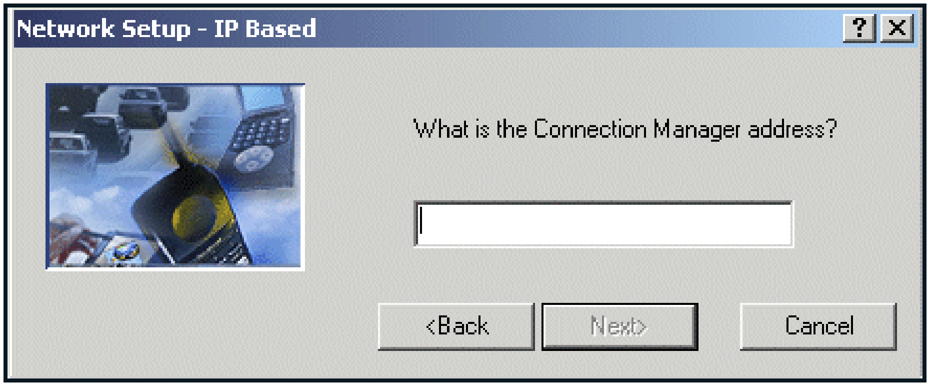
Select the network type(s) this new connection will use. Make sure all modems and network adapters are available before continuing.

IP, WiFi, GPRS, 1xRTT, Broadband
DATARADIO
DataTAC and Private Mobile Radio
Dial
Mobitex
Norcom Satellite

Figure 10-4 Selecting a network

Next, provide the network IP address of the Connection Manager.

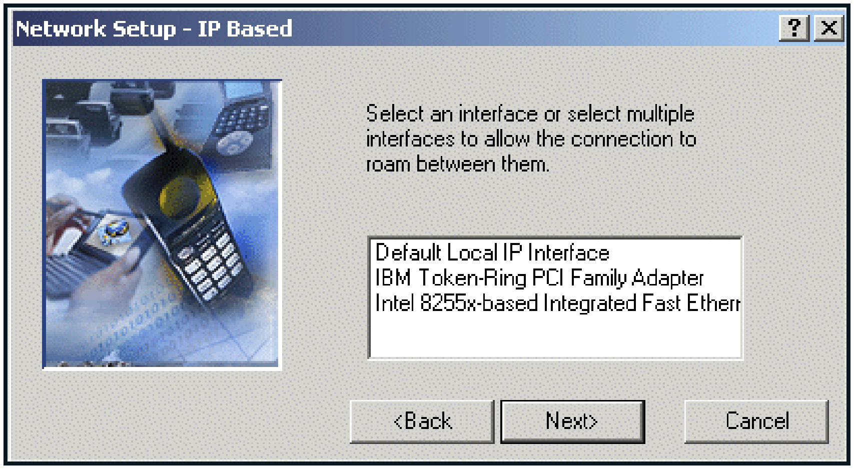
Note: Make sure all modems and network adapters are available before continuing with the creation of a network connection from this point.


Network Setup - IP Based

What is the Connection Manager address?

Figure 10-5 Connection Manager IP address

After providing the IP address of the Connection Manager, you will be prompted to select an interface or multiple interfaces to allow the connection to roam between them.



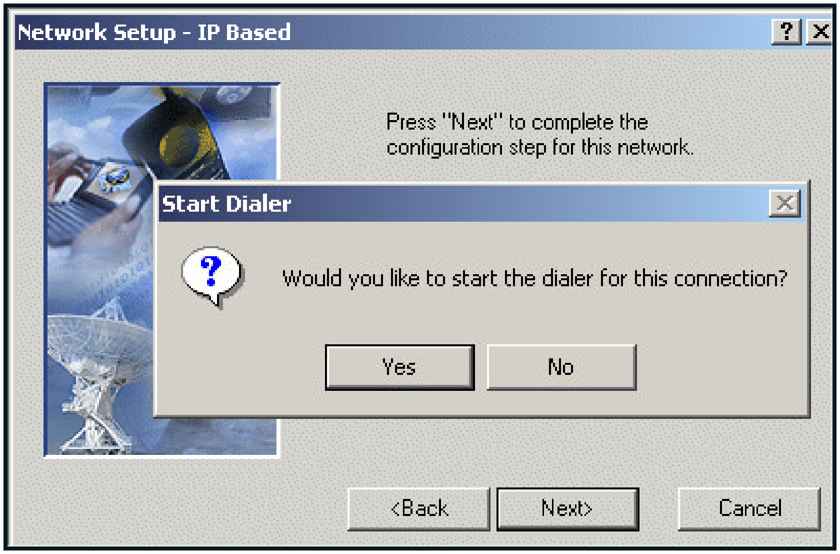
Network Setup - IP Based

Select an interface or select multiple interfaces to allow the connection to roam between them.

Default Local IP Interface
IBM Token-Ring PCI Family Adapter
Intel 8255x-based Integrated Fast Etherr

Figure 10-6 Select interface or multiple interfaces

Once the interface is selected, the wizard will complete configuring your connection.



Network Setup - IP Based

Press "Next" to complete the configuration step for this network.

Start Dialer

Would you like to start the dialer for this connection?

Figure 10-7 Network setup

Once the wizard is completed, your new Mobility Client connection should appear as illustrated in Figure 10-8.

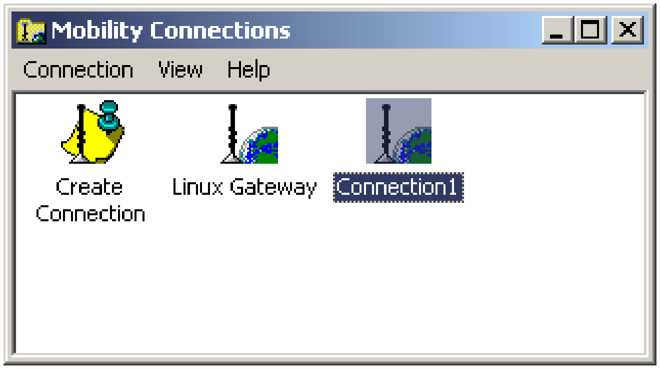
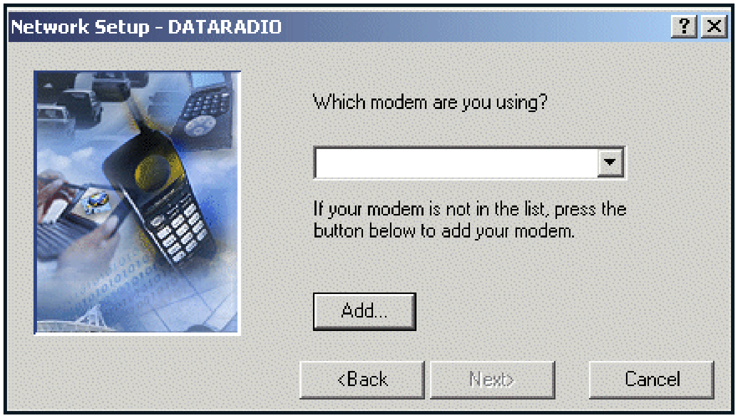


Figure 10-8 Mobility connections

If you are configuring a non-IP based connection, the following wizard panels will appear once you have completed the step for adding the IP address of the Connection Manager.



Which modem are you using?

If your modem is not in the list, press the button below to add your modem.

Figure 10-9 Modem selection

Once the Network Setup panel appears, use the pull-down to select your modem. Otherwise, select the **Add** button, and add or select the appropriate modem you have installed. Once the interface has been selected, the wizard will complete configuring your connection. See Figure 10-10.

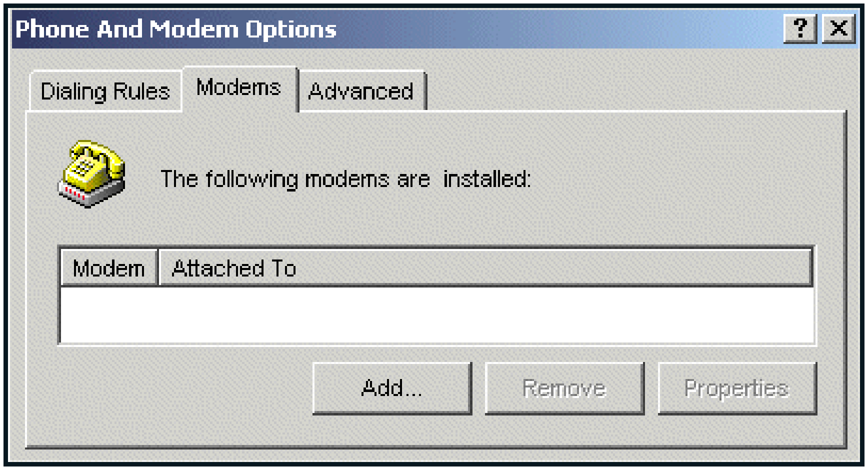


Figure 10-10 Phone and modem options

10.7 Configuring a connection

Once the Mobility Client for you connection has been created, you can now make changes to the connection by right-mouse clicking the existing configuration icon. A pop-up menu for that connection will appear. Select **Properties** to examine and modify the connection properties.

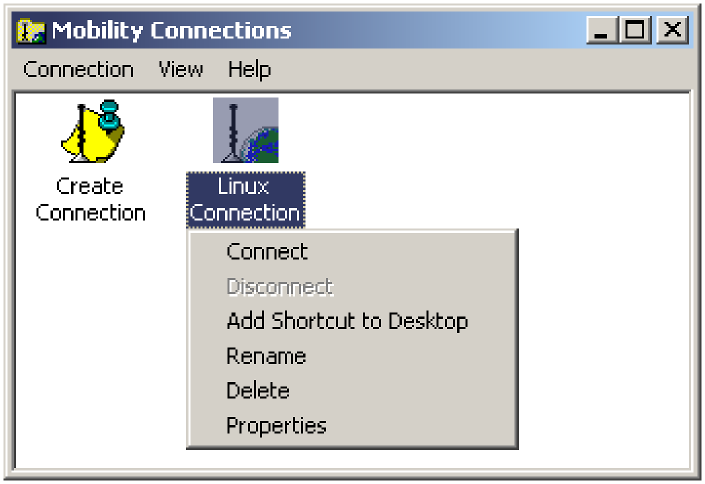
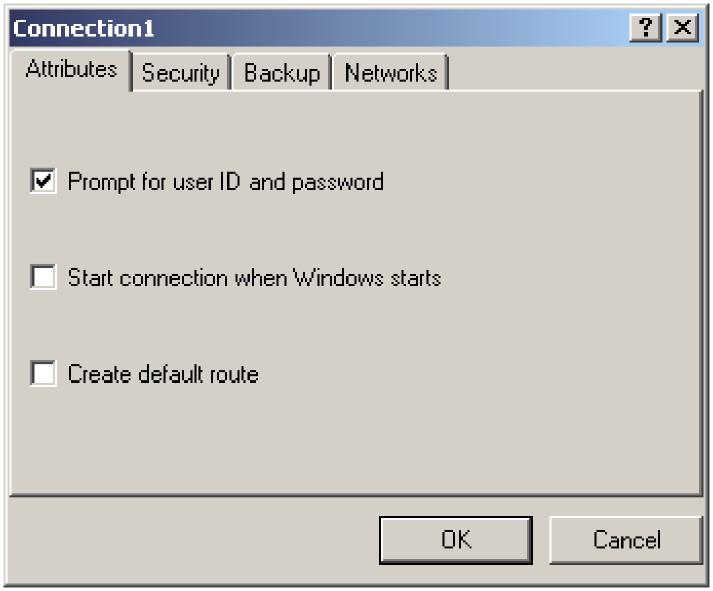


Figure 10-11 Mobility connections

The Properties screen has a series of tabs. On the **Attributes** tab you can change security options regarding user ID and password. If you deselect Prompt for User ID and Password, you will not be prompted to enter this information each time the connection is started. You can select the option for starting the connection when Windows starts.

Another option here is to create a default route for the traffic. If **Create Default route** is selected, then the Mobility Client will add an entry in the local routing table that causes all IP traffic that cannot find a specific route to be routed over the Mobility Client interface. This is of particular interest for clients that are on a multihomed system.



☑ Prompt for user ID and password

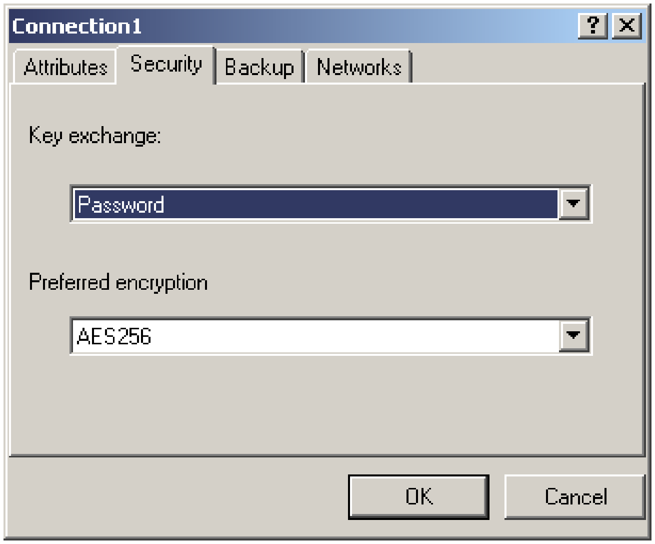
□ Start connection when Windows starts

□ Create default route

Figure 10-12 Attributes

Note: The Create default route option is not available on Windows CE.

The second tab is the Security Property tab. The Security tab provides options for changing the security option regarding the password's key exchange and preferred encryption.

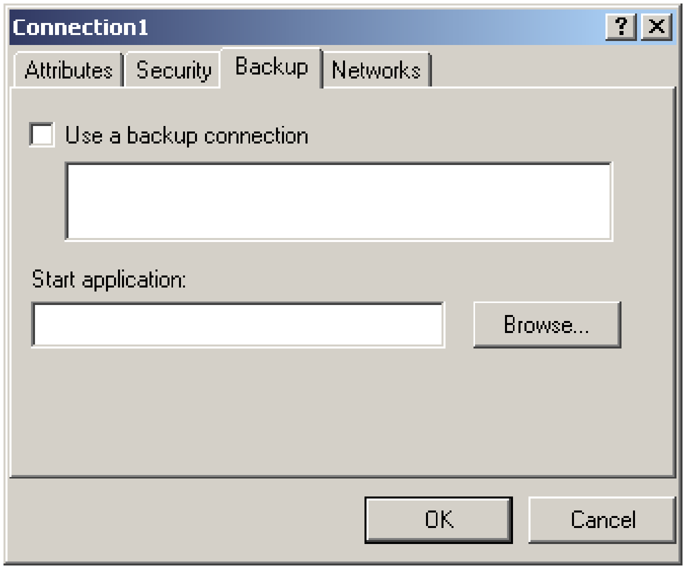


Key exchange: Password

Preferred encryption AES256

Figure 10-13 Security

The next tab on the Properties screen is the **Backup** tab. This tab is for defining a backup connection. This panel is used to define a backup connection. It also configures a start application to run programs when a connection is established with the Connection Manager. See Figure 10-14.

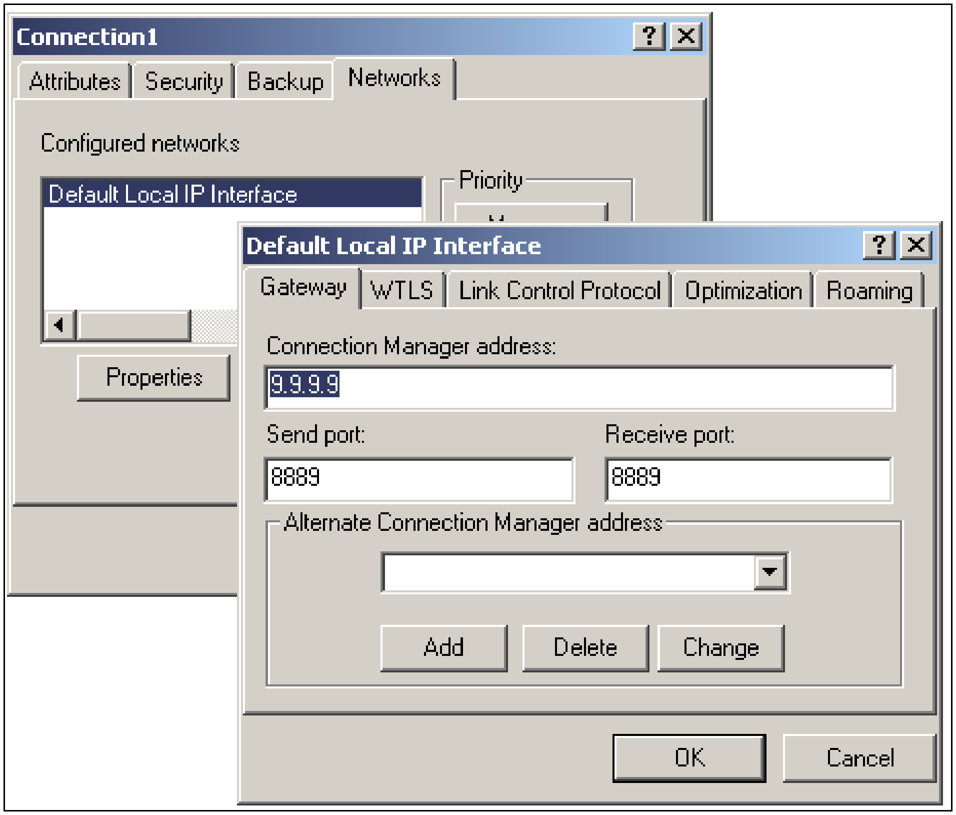


□ Use a backup connection

Start application:

Figure 10-14 Backup

The next tab on the Property screen is the **Networks** tab. The Network screen is used to configure your Mobility Client network. Based on your configured network, you can select that network and determine other properties for that network. See Figure 10-15.



Configured networks: Default Local IP Interface

Connection Manager address: 9.9.9.9

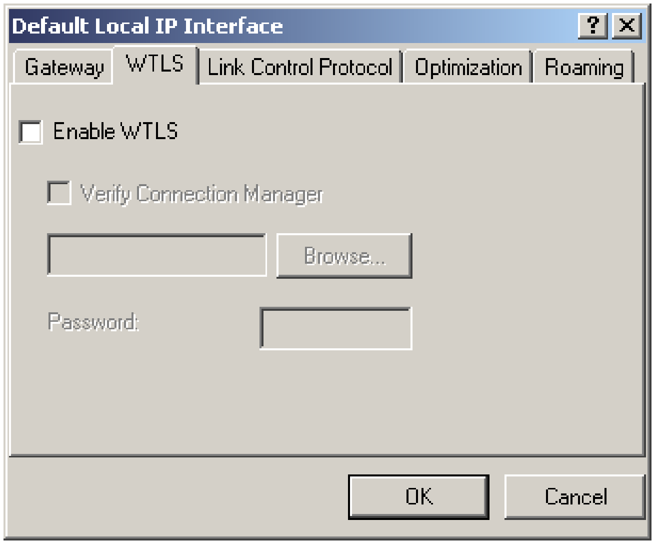
Send port: 8889

Receive port: 8889

Figure 10-15 Networks

When selecting the **Network Properties** button, the Default Interface menu will appeared. The initial tab of the Default Properties option is the Gateway tab. The Gateway tab allows you to change the gateway IP address, and the send and receive ports for communications. The panel also allows you to add, delete, and modify alternate Connection Manager addresses for the default network connection selected.

The next panel is the Wireless Transport Layer Security (WTLS) property options for Mobility Clients. For more information about WTLS for Mobility Clients, see Chapter 12, “Mobility Client security” on page 211.



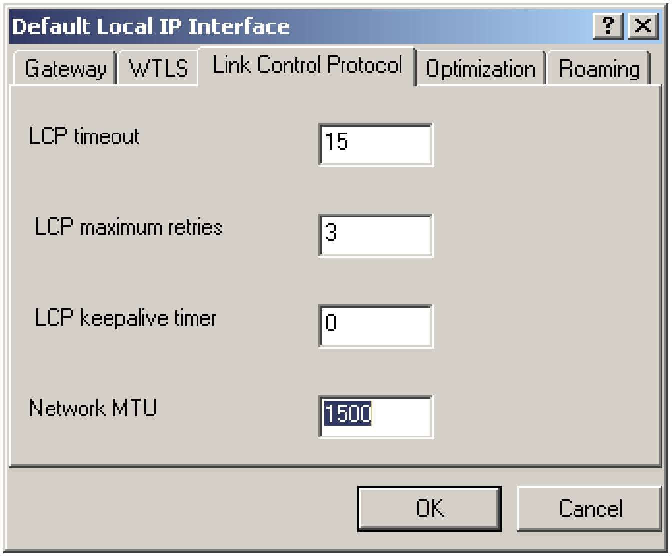
□ Enable WTLS

□ Verify Connection Manager

Password:

Figure 10-16 Wireless Transport Layer Security Property (WTLS)

The next tab in Network Properties is the **Link Control Protocol** tab, which specifies the timer values for your connections. Also see Chapter 13, “MQe application traffic optimization” on page 247.



LCP timeout 15

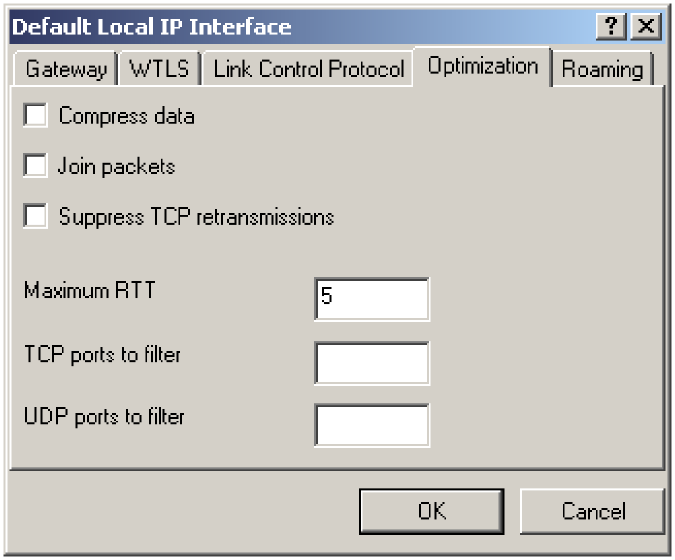
LCP maximum retries 3

LCP keepalive timer 0

Network MTU 1500

Figure 10-17 Link Control Protocol timer values

The next tab is the Optimization tab for Network Properties as shown in Figure 10-18.



Default Local IP Interface

□ Compress data

□ Join packets

□ Suppress TCP retransmissions

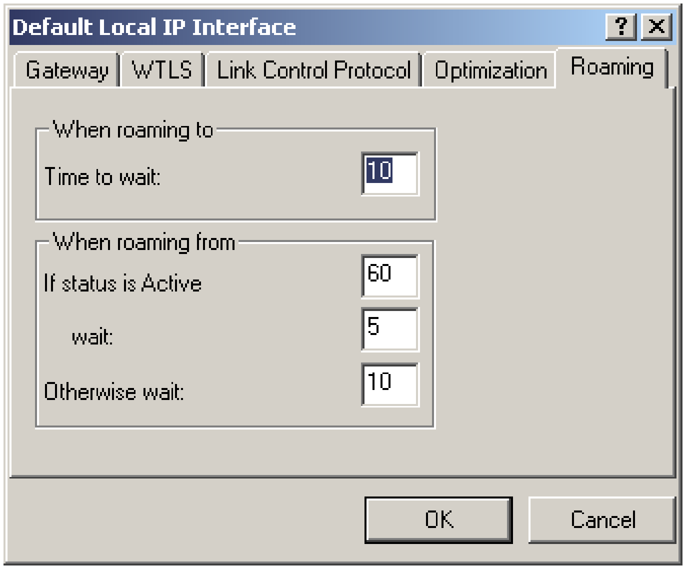
Maximum RTT: 5

TCP ports to filter:

UDP ports to filter:

Figure 10-18 Optimization parameters

For more information about roaming, see Chapter 21, “Roaming” on page 457.



Default Local IP Interface

When roaming to

Time to wait: 10

When roaming from

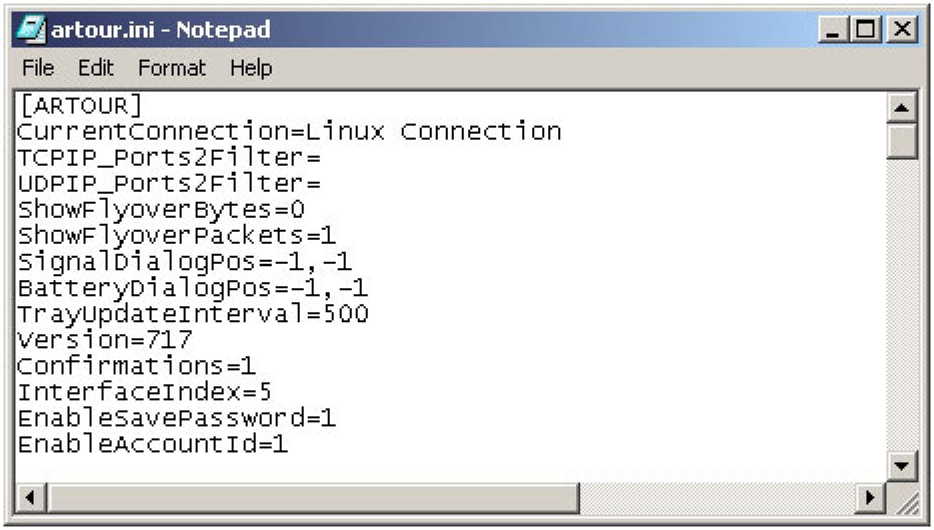
If status is Active: 60

wait: 5

Otherwise wait: 10

Figure 10-19 Roaming

All of the configuration data specified using the Configuration Wizard or the Properties menu is stored in a file named *Artour.ini*. For example, see Figure 10-20.



```
[ARTOUR]
CurrentConnection=Linux Connection
TCPIP_Ports2Filter=
UDPIP_Ports2Filter=
ShowFlyoverBytes=0
ShowFlyoverPackets=1
SignalDialogPos=-1,-1
BatteryDialogPos=-1,-1
TrayUpdateInterval=500
Version=717
Confirmations=1
InterfaceIndex=5
EnableSavePassword=1
EnableAccountId=1
```

Figure 10-20 Mobility Client configured parameters

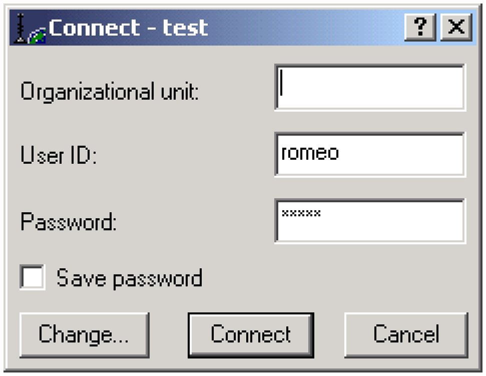
The following are some recommendations to properly configure the Mobility Client:

- Be very careful about editing this file because unexpected results may occur if errors are made.
- The artour.ini file has many more configurable options than displayed through the graphical user interface (GUI).
- A description of all of the possible configuration settings can be found in the *Connection Manager, Version 5, Administration Guide.*
- Do not edit the autour.ini file while any Mobility Client programs are running or your changes may not be saved properly.

10.8 Connecting to the gateway

In the Connection Window, when you double-click the connection in this window you will be provided with the Connect panel. Depending on the type of network connection, you will see one of the two Connect screens. The main difference is whether you have a Change button to change phone and location settings. The dial or remote networking connection has this option. You must enter an

Organization Unit, a user ID, and password depending on how the gateway and the client were configured. The system will also remember the password so you do not have to enter it each time.



Organizational unit:

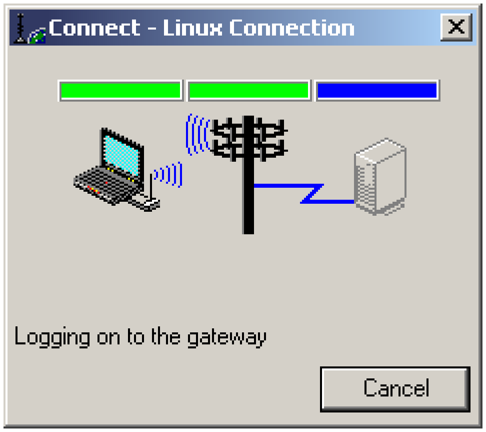
User ID: romeo

Password: *****

□ Save password

Figure 10-21 Connecting to Connection Manager gateway

As illustrated in Figure 10-22, there are three stages to the connection process. A graphical message box will appear and provide you progress. The first step illustrates opening the network device (modem, device card, etc.). The second step provides the status on connecting to the network (for example, range indicator from the tower). Lastly, is the exchanging credentials and capabilities with the gateway (minimum of four packets exchanged, two sent and two received).



Logging on to the gateway

Figure 10-22 Connecting to Connection Manager Mobility Client gateway

The client program is represented by the atennae/world icon in the Windows tray as illustrated in Figure 10-23.

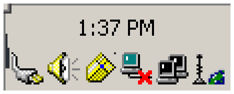


Figure 10-23 Mobility Client icon

After the gateway connects, you can monitor the status of your connection by examining the Windows tray icon. Whenever data is being exchanged, the icon in the tray shows a lighting bolt. In order to get additional information, move your mouse pointer over the icon. Fly-over text will indicate the amount of information being processed.

If you right-mouse click the **system tray** icon, you will receive an operations pop-up menu as shown in Figure 10-24.

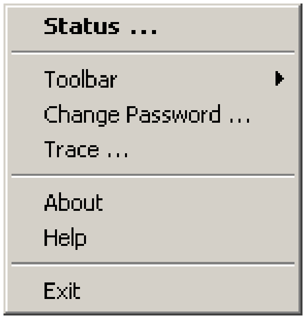


Figure 10-24 Operations pop-up menu

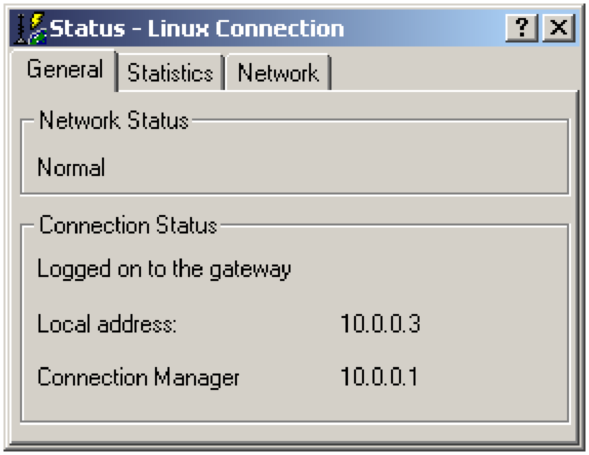
This menu provides you a number of options including Status (which brings up status dialogs). Additional operations available on the Operation pop-up menu include:

- Toolbar - To show or automatically display the Mobility Client tool bar
- Change Password allows you to change your Connection Manager password.
- Trace activates the trace setting program.
- About displays the current version of the Mobility Client.
- Disconnect closes your gateway connection and exits the Mobility Client program.

Status indications

If you right-mouse click the **tray icon** and select **Status** from the Operations pop-up menu, you will see the Status window. The Status window has the following tabs:

1. The General tab displays the network and connection status as illustrated in Figure 10-25.



Network Status

Normal

Connection Status

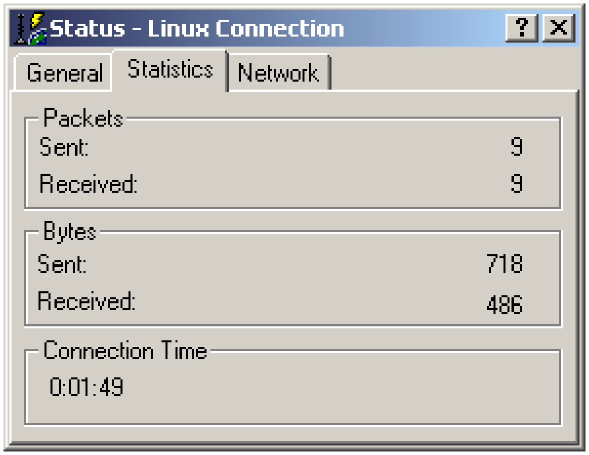
Logged on to the gateway

Local address: 10.0.0.3

Connection Manager 10.0.0.1

Figure 10-25 General

2. The Statistics tab shows packets, bytes, and the duration of the connection as illustrated in Figure 10-26.



Packets

Sent: 9

Received: 9

Bytes

Sent: 718

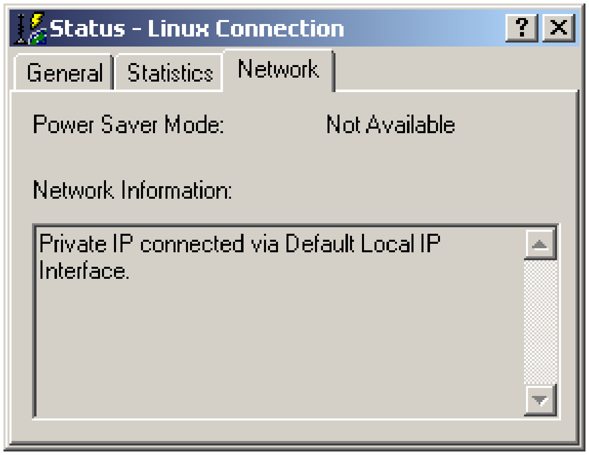
Received: 486

Connection Time

0:01:49

Figure 10-26 Statistics

The Network tab shows the modem or device information as illustrated in Figure 10-27.



Power Saver Mode: Not Available

Network Information:

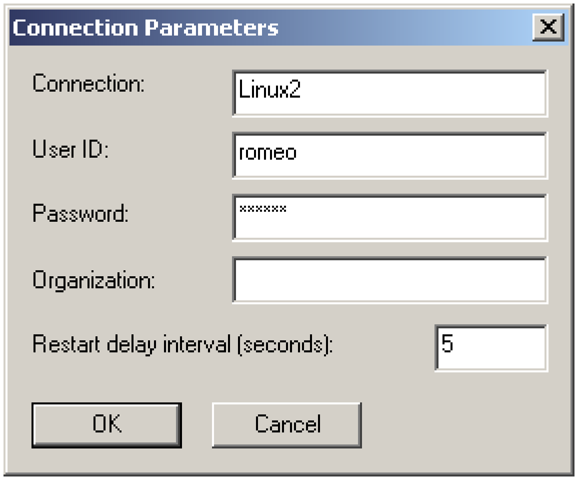
Private IP connected via Default Local IP Interface.

Figure 10-27 Network

10.9 Mobility Client toolkit and example

The Mobility Client toolkit and Application Program Interfaces (APIs) are provided with the Mobility Client in order for the user to create network-aware applications. One type of application, for example, selects the exact type of data transmit based on the type of connectivity, cost, and bandwidth. Another application can monitor Wi-Fi signal strength, decide to start a General Packet Radio Switching (GPRS) connection, and roam to GPRS before Wi-Fi connectivity is lost.

The Client Monitor program, monitor.exe is located with the Mobility Client toolkit. This example can be used to maintain the connection to the Connection Manager even when errors occur. This is useful when there is not an operator available to monitor the connection status. In running the example for the first time, the monitor will not be able to locate the monitor configuration file, monitor.ini, and will prompt you for the configuration information shown in Figure 10-28.



Connection: Linux2

User ID: romeo

Password: ******

Organization:

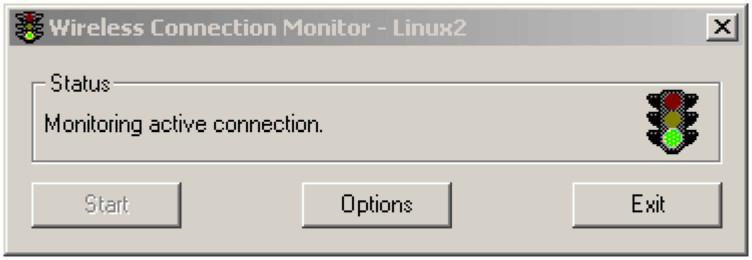
Restart delay interval (seconds): 5

Figure 10-28 Connection parameters

The connection parameters are:

- **Connection**: Required. The connection that you created as part of the Mobility Client configuration.
- **User ID**: Optional. If your connection requires a user ID, enter it here.
- **Password**: Optional. If your connection requires a password, enter it here.
- **Organization**: Optional. If your connection requires an organization qualifier, enter it here.
- **Restart Delay**: Required. Enter the time (in seconds) that the monitor waits to restart the Mobility Client after it detects an error.

When the monitor program is started, it checks to see if the Mobility Client connection is active and logged on to the Connection Manager. If it is not, it starts the Mobility Client using the configuration connection profile. Once the connection is active and logged on, it then waits for disconnect events from the Mobility Client.

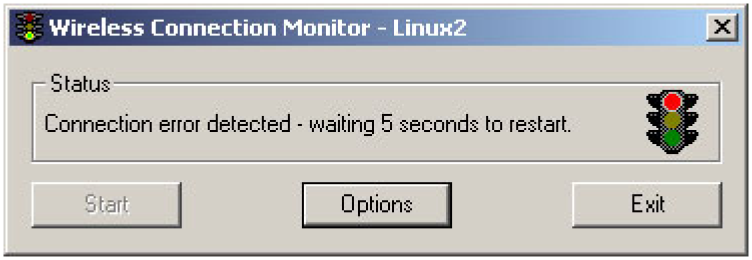

Wireless Connection Monitor - Linux2

Status

Monitoring active connection.

Figure 10-29 Status

If an event is received, the monitor programs waits for the number of seconds specified for the Restart Delay parameter. See Figure 10-30.

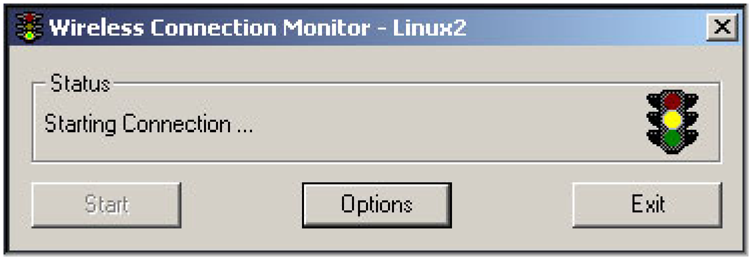

Wireless Connection Monitor - Linux2

Status

Connection error detected - waiting 5 seconds to restart.

Figure 10-30 Connection error detected

Next, the monitor starts the Mobility Client again as shown in Figure 10-31.


Wireless Connection Monitor - Linux2

Status

Starting Connection ...

Figure 10-31 Restarting the connection

For additional development examples and API information refer to the Mobility Client toolkit.

11

Mobility Client connectivity

This chapter describes the IBM Mobility Client connectivity and how it works in a WebSphere Everyplace Connection Manager environment. The topics covered in this chapter include:

- Networking
- Data flow
- Packets flow
- Headers

11.1 Networking

Any device must have at least one adapter to get network connectivity. This adapter must be able to communicate with the network standard. For example, an Ethernet adapter is able to connect only with Ethernet networks.

Once the adapter is able to connect the network, the connection must be established. This happens as soon as the adapter validates and receives an IP address, and validates and receives a default gateway. This IP address must be part of the subnet where the adapter is connected. It will be called “Real IP”.

At the moment when the IBM Mobility Client becomes active, the WebSphere Everyplace Connection Manager gives the client one IP number that is called “Virtual IP”.

Initial considerations

The most common data networks used today are IP based, so the concepts are focused in this kind of network. The following resources are used in IP data networks:

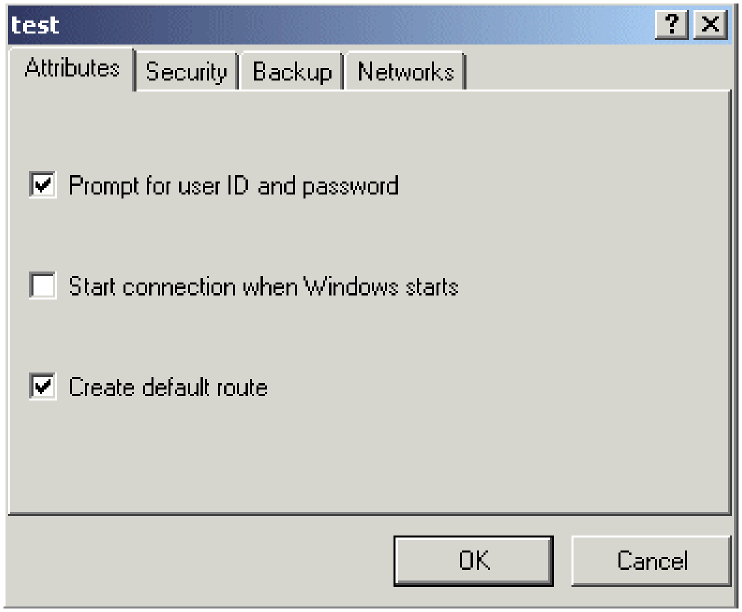
- **IP**: Internet protocol. Is the address for each network interface adapter connected to an IP network. This address can be predefined or can be delivered by a DHCP server.
- **DHCP:** Dynamic Host Configuration Protocol is a protocol that enables a DHCP server to assign IP addresses to an TCP/IP adapter.
- **Default Gateway:** This is the IP address that handles the communication between two machines from a different subnet. Usually it is an IP address of a router between these two subnets. If this router is not able to find the other machine, it pushes the packet to the one router that can find it.
- **Subnet Mask**: Distinguishes the network ID from the host ID. This is used to specify if the host is local or remote. This mask is the way to describe how much of the address is to the network, and how much is for the host part.

11.2 Data flow

Data flows in three different ways depending on the configuration and state of the client. For example:

1. IBM Mobility Client inactive
2. IBM Mobility Client active and the option Create Default Route is disabled.
3. IBM Mobility Client active and the option Create Default Route is enabled.

The option Create Default Route is in the client's attributes shown in Figure 11-1. Furthermore, this will describe how the data flows in these three situations.



☑ Prompt for user ID and password

□ Start connection when Windows starts

☑ Create default route

Figure 11-1 - Connection Manager Client - Attributes

To better understand the reasons of the flow, some understanding of routes must be acquired so it will be described with any of these situations.

All the examples used in this chapter were obtained from the host used for the IBM Mobility Client installation and testing during the production of this chapter.

IBM Mobility Client inactive

This is the usual connection that any device with a network adapter has with the network. In this situation, the IBM Mobility Client is not controlling the data flow.

To understand it better the routes must be analyzed. To see the routes in a Windows environment (for example, at the prompt) the command is **`route print`** as in Example 11-1.

This first example was obtained with the IBM Mobility Client inactive.

Example 11-1 Route Print - IBM Mobility Client inactive

```
C:\>route print
===========================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x1000003 ...00 60 94 89 dd c9 ...... IBM Token-Ring PCI Family Adapter
0x1000004 ...00 09 6b 64 9d a2 ...... Intel 8255x-based Integrated Fast
Ethernet

0x1000005 ...00 06 13 19 58 00 ...... IBM Mobility Network Interface
===========================================================================
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0       9.24.104.1    9.24.104.172       1
       9.24.104.0    255.255.254.0     9.24.104.172    9.24.104.172       1
     9.24.104.172  255.255.255.255        127.0.0.1       127.0.0.1       1
    9.255.255.255  255.255.255.255     9.24.104.172    9.24.104.172       1
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1       1
        224.0.0.0        224.0.0.0     9.24.104.172    9.24.104.172       1
  255.255.255.255  255.255.255.255     9.24.104.172         1000003       1
Default Gateway:        9.24.104.1
===========================================================================
Persistent Routes:
  None
```

The first column of the active route table displays the network destination. When a packet is to be routed, the destination address of the packet is compared to those listed in the route table using the destination and netmask combination. When a valid route is found, the packet is forwarded through the corresponding gateway and interface. For example, if a packet is sent to address 9.24.104.25, a suitable route is found in the routing table from the entry listing 9.24.104.0 in row 2 in Example 11-1. The packet will now be sent to its destination by the interface IP 9.24.104.172. This happens because the destination address is inside the same subnet so the gateway is the host IP itself.

As another example, if a packet is destined for address 10.0.123.200, a suitable host is found in the routing table from the entry listing 0.0.0.0 in row 1 in Example 11-1. The packet will now be sent to its destination by the interface IP 9.24.104.172 using the gateway 9.1.104.1. This happens because the destination address is outside the host’s interface subnet so the gateway is the default gateway of the subnet.

In the host used to produce this chapter, the active interface 9.24.104.172 is the Ethernet adapter IP address shown in Example 11-2. The other adapter is inactive as `Cable Disconnected`.

Example 11-2 - Network interfaces

```
C:\>ipconfig

Windows 2000 IP Configuration

Ethernet adapter {BF31B7C0-1A19-469E-A562-CAB9F0D73EA0}:

        Media State . . . . . . . . . . . : Cable Disconnected

Ethernet adapter Local Area Connection 2:

        Connection-specific DNS Suffix  . : itso.ral.ibm.com
        IP Address. . . . . . . . . . . . : 9.24.104.172
        Subnet Mask . . . . . . . . . . . : 255.255.254.0
        Default Gateway . . . . . . . . . : 9.24.104.1
```

So, the conclusion is that all packets that flow between the host and network are forwarded to the Ethernet adapter. In this Example 11-1, the IBM Mobility Client is not active. This flow looks like the Figure 11-2.

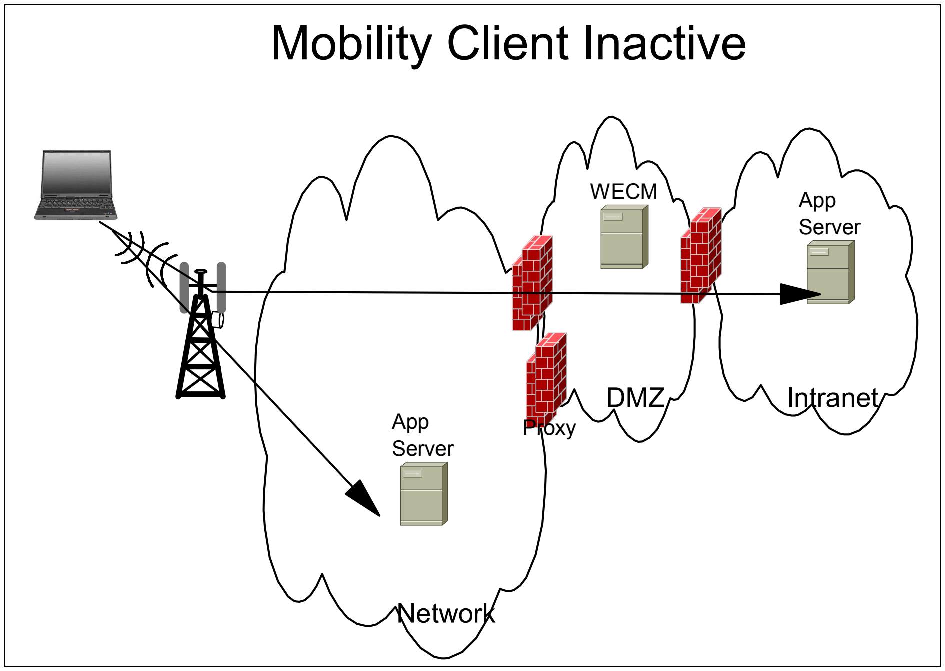


Figure 11-2 Mobility Client inactive - Network view

Mobility Client active and Create Default Route option disabled

In this situation, the IBM Mobility Client controls the data flow between the device and WebSphere Everyplace Connection Manager local network, for example, the private intranet. The packets flowing outside this network, for example, public Internet will flow directly by the network adapter and will not be controlled.

To understand it better, will be used the same resource used before to display the routes in the first situation. Some differences can be easily found in this new route table as in Example 11-3.

Example 11-3 IBM Mobility Client active - Create Default Route option disabled

```
C:\>route print
============================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x1000003 ...00 60 94 89 dd c9 ...... IBM Token-Ring PCI Family Adapter
0x1000004 ...00 09 6b 64 9d a2 ...... Intel 8255x-based Integrated Fast Etherne

0x1000005 ...00 06 13 19 58 00 ...... IBM Mobility Network Interface
============================================================================
============================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0       9.24.104.1     9.24.104.172       1
       9.24.104.0    255.255.254.0     9.24.104.172     9.24.104.172       1
     9.24.104.172  255.255.255.255        127.0.0.1        127.0.0.1       1
    9.255.255.255  255.255.255.255     9.24.104.172     9.24.104.172       1
       10.0.123.0    255.255.255.0       10.0.123.5       10.0.123.5       1
       10.0.123.5  255.255.255.255        127.0.0.1        127.0.0.1       1
   10.255.255.255  255.255.255.255       10.0.123.5       10.0.123.5       1
        127.0.0.0        255.0.0.0        127.0.0.1        127.0.0.1       1
        224.0.0.0        224.0.0.0     9.24.104.172     9.24.104.172       1
        224.0.0.0        224.0.0.0       10.0.123.5       10.0.123.5       1
  255.255.255.255  255.255.255.255       10.0.123.5          1000003       1
Default Gateway:        9.24.104.1
============================================================================
Persistent Routes:
  None
```

In this situation, if a packet is destined for address 9.24.104.25, the suitable route keeps the same as found in Example 11-1 on page 194. The packet will still be sent to its destination by the interface IP 9.24.104.172, but in the second example used, with a packet destined for address 10.0.123.200, the suitable route is different in the IBM Mobility Client active system. This happens because the IBM Mobility Client is working on the TCP/IP stack, and adds the default route for its own subnet. So, the packet destined for address 10.0.123.200 is sent to its destination by the interface 10.0.123.5 as found in row 5 in Example 11-3. This is the interface that is recognized as the virtual IP described before.

This connection, with the IBM Mobility Client active, drives to the client interface and then to WebSphere Everyplace Connection Manager server just the packets destined to client's subnet or server's intranet. The other packets destined to the other subnet such as the public Internet is sent by the interface IP of the adapter. In this situation, only the packets forwarded by the IBM Mobility Client can be compressed, roamed, and protected by the client's VPN.

The host with IBM Mobility Client active now has two adapters active as in Example 11-4. The first adapter is the virtual adapter provided by the IBM Mobility Client driver. The second adapter is the physical adapter that was active at the first example.

Example 11-4 - Network interfaces

```
C:\>ipconfig

Windows 2000 IP Configuration

Ethernet adapter {BF31B7C0-1A19-469E-A562-CAB9F0D73EA0}:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 10.0.123.5
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . :

Ethernet adapter Local Area Connection 2:

        Connection-specific DNS Suffix  . : itso.ral.ibm.com
        IP Address. . . . . . . . . . . . : 9.24.104.172
        Subnet Mask . . . . . . . . . . . : 255.255.254.0
        Default Gateway . . . . . . . . . : 9.24.104.1
```

The view of this type of connection is shown Figure 11-3.

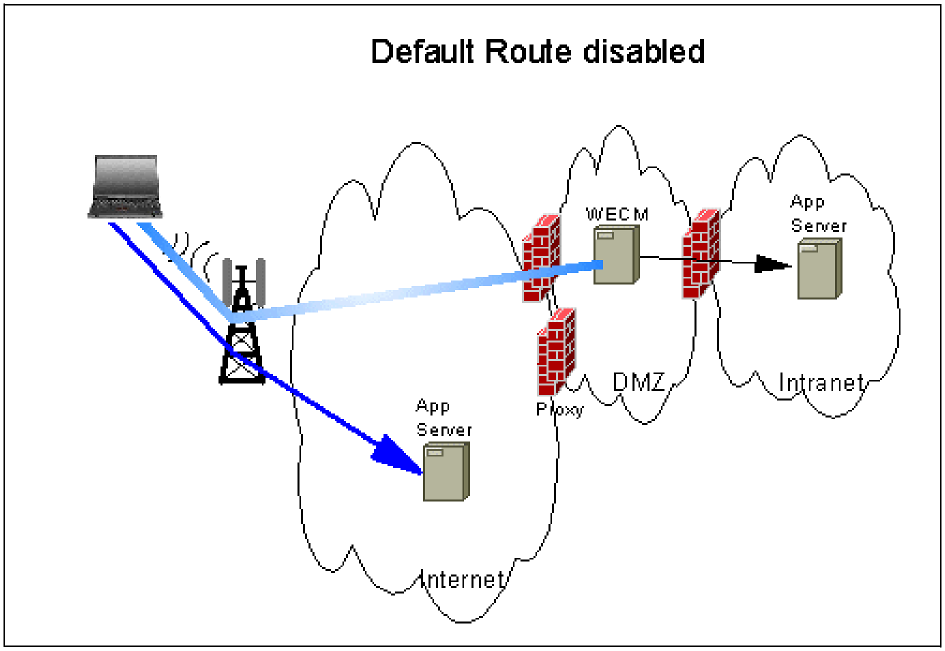


Figure 11-3 Mobility Client active - Default route disabled - Network View

Mobility Client active and Create Default Route option enabled

In this situation, the IBM Mobility Client will control all data flow between the device and network, private intranet, or public Internet.

For example, when using the same resource to display routes, some differences can be seen in this routing table as illustrated in Example 11-5.

Example 11-5 Route print: IBM Mobility Client active Create default route option enabled

```
C:\>route print
===========================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x1000003 ...00 60 94 89 dd c9 ...... IBM Token-Ring PCI Family Adapter
0x1000004 ...00 09 6b 64 9d a2 ...... Intel 8255x-based Integrated Fast
Ethernet

0x1000005 ...00 06 13 19 58 00 ...... IBM Mobility Network Interface
===========================================================================
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0       10.0.123.1     10.0.123.5       1
```

```
     9.24.104.172  255.255.255.255        127.0.0.1       127.0.0.1       1
    9.255.255.255  255.255.255.255     9.24.104.172    9.24.104.172       1
       10.0.123.0    255.255.255.0       10.0.123.5      10.0.123.5       1
       10.0.123.5  255.255.255.255        127.0.0.1       127.0.0.1       1
   10.255.255.255  255.255.255.255       10.0.123.5      10.0.123.5       1
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1       1
        224.0.0.0        224.0.0.0       10.0.123.5      10.0.123.5       1
  255.255.255.255  255.255.255.255       10.0.123.5         1000003       1
Default Gateway:        10.0.123.1
===========================================================================
Persistent Routes:
  None
```

This route table displays that if a packet is destined for address 9.24.104.25, a suitable route is found in the routing table from the entry listing 0.0.0.0 in row 1 in Example 11-5. The packet will now be sent to its destination by the interface IP 10.0.123.5 through gateway 10.0.123.1. This gateway IP address is the virtual IP address of the WebSphere Everyplace Connection Manager. If the packet is destined for address 10.0.123.200, the packet will be sent to its destination by the interface IP 10.0.123.5 though gateway 10.0.123.1. This means that all packets in this third situation pass through the IBM Mobility Client, and are forwarded to the IBM WebSphere Everyplace Connection Manager as in Figure 11-4.

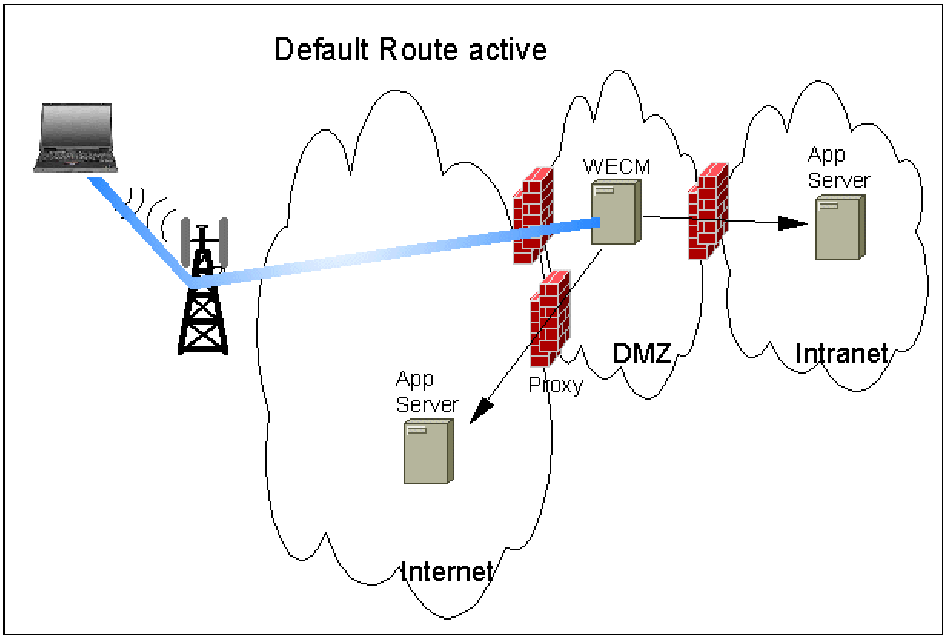


Figure 11-4 Mobility Client active - Default route enabled - Network view

With this last example, all the communication between the device and network is controlled by the IBM Mobility Client and IBM WebSphere Everyplace Connection Manager, so the features provided by the solution should work in all packet flows. Once the IBM Mobility Client is active and the option **Create Default route** enabled, the client works in the stack TCP defining routes that make the packets flow through the client driver and then through the WebSphere Everyplace Connection Manager server. In this situation, all the communication that the device makes with the network will be controlled by the IBM Mobility Client. This control includes the features selected as security, compression, roaming, etc.

11.3 Packet flow

Keeping in mind the concepts described in 11.2, “Data flow” on page 192, and 11.3, “Packet flow” on page 200, the session will describe how the packets flow since its source application to the destination application, and also the return of the answer of this packet.

This flow can occur in three different ways depending on the configuration. This also depends on the client status and the availability of the Create Default Route. So again, we will describe each of the possible situations.

IBM Mobility Client inactive

As explained in data flow section, the data in this situation flows directly by the physical network adapter. This flow is explained in Figure 11-5.

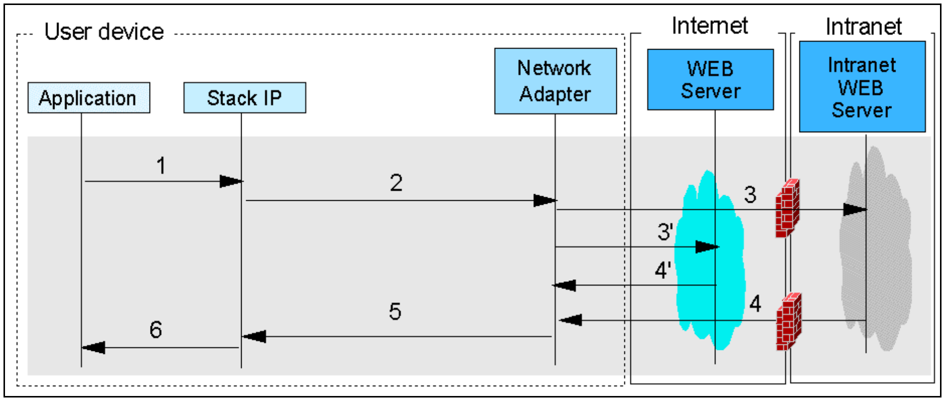


Figure 11-5 Packet Flow - Mobility Client inactive

This is the usual connection network. In this kind of connection the packets are not secured by a VPN tunnel and they are exposed to attack. For example, as illustrated in Figure 11-5:

1. The application sends data to communicate with the network.
2. The IP receives the information and sends it to the Physical Network Adapter with the necessary headers so the packet can be addressed to the network.
3. The Physical Network Adapter sends this packet to the Internet routers to reach the correct destination inside the enterprise intranet. This can only happen if the firewall is set to accept this connection, and the static route is enabled so the packet can reach the selected IP range. When access is for the Internet, the Physical Network Adapter sends the packet to the Internet routers to reach the correct destination on the Internet.
4. The application server sends the response through the firewall, passing through the Internet to the adapter. Alternatively, when access is for the Internet, the application server sends the response to the adapter.
5. The adapter delivers the packets to the IP stack, and the headers will be removed.
6. The adapter sends the information to the application.

Mobility Client active and Create Default Route option disabled

In this situation, the access will be partially protected. As explained before, the data flow will now depend of the kind of access that the application and user intend to do. If the application goes to the public or local network, the packets will be delivered without VPN or the IBM Mobility Client interference. By the way, if the application or user intends to access intranet application servers, the packets will be delivered to IBM Mobility Client, and therefore receive all the benefits provided by the tool such as compression, encryption, VPN, roaming, etc. The packet’s flow is in Figure 11-6.

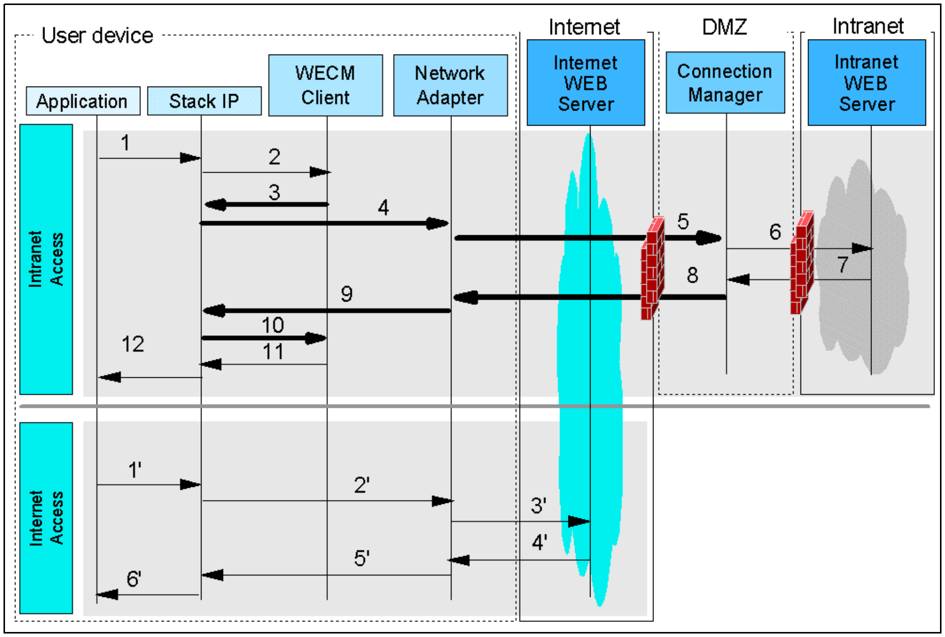


Figure 11-6 Packet Flow - Mobility Client active - Default route disabled

This is the most common situation used by the enterprises. The access to the intranet applications is encapsulated by a VPN solution, and the access to the Internet applications is independent of the client. In any case, this second access is not protected by a VPN. This model is used by the enterprises that wants to provide remote access to employees, business partners, and others, and does not want to increase by too much the data flow inside the intranet. In addition, the user device cannot provide additional protection to the Internet data.

The packet flow in this section will be described in one phase only, even though there exists two different phases:

- The initial phase is described in this section and refers to the intranet access.
- The second phase refers to the Internet access, and the concept is exactly as shown in Figure 11-5.

Therefore, (as illustrated in Figure 11-6) the intranet access packet flow is as follows:

1. The application requests network access to the IP stack. The stack is routed to send all requests to the IBM Mobility Client (more informations is in “Mobility Client active and Create Default Route option disabled” on page 196).

2. The IP stack sends the request to the IBM Mobility Client, and the client starts to execute the services selected such as data and header compresses, encryption, etc., and then builds the headers to deliver the packet.
3. IBM Mobility Client sends the packets back to the IP stack.
4. The IP stack forwards it to the network interface adapter.
5. The adapter sends the packets to the network addressed to IBM WebSphere Everyplace Connection Manager server.
6. IBM WebSphere Everyplace Connection Manager receives the packets, decrypts it, decompresses, etc., and sends the packets to the target intranet application server using the virtual IP as the source IP.
7. The application server answers the request and sends it to the IBM Mobility Client, but the network is prepared to deliver these packets to the WebSphere Everyplace Connection Manager server.
8. The Connection Manager server encrypt, compress, etc., and sends the packets back to the device network adapter.
9. The device network adapter sends the data to the IP stack.
10. The IP stack sends it to IBM Mobility Client.
11. IBM Mobility Client decompresses, decrypts, etc. the packet and then delivers it to the IP stack.
12. The IP stack sends it to the application.

Mobility Client active and Create Default Route option enabled

This scenario is the most secure for the user’s data flow. In this situation all packets are delivered to the IBM WebSphere Everyplace Connection Manager, and can be protected by the VPN. Also, you can get the advantage of compression, encryption, etc. It occurs independently of the destination of the packet, intranet, or public/local network. This flow is illustrated in Figure 11-7.

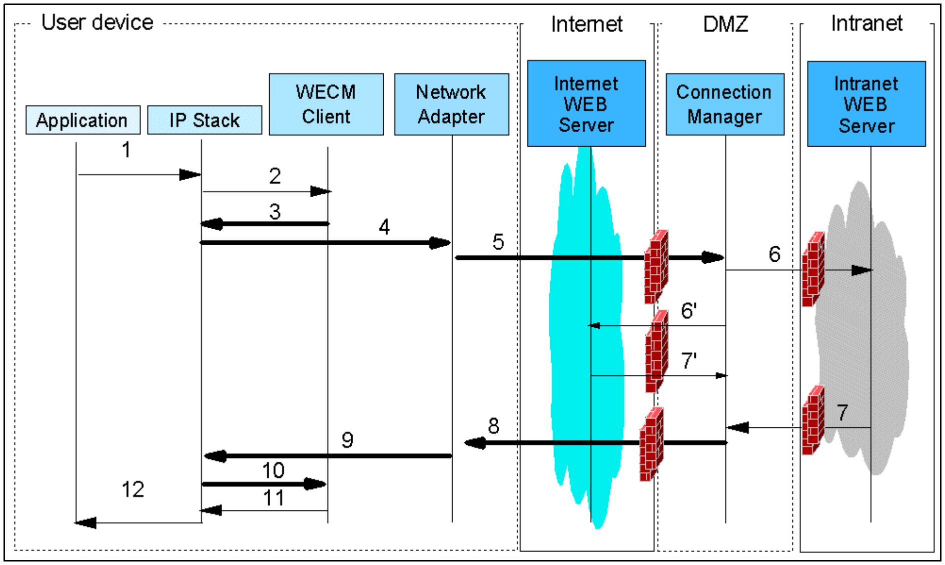


Figure 11-7 Packet flow - Mobility Client active - Default route enabled

This scenario is used by enterprises and telcos that want to keep packet flow information for all the users. However, this option increases significantly the volume of data flowing in the intranet. The explanation of the steps of this flow is as follows:

1. The application requests network communication to the IP stack.
2. The IP stack sends the request to IBM Mobility Client, and the client starts to execute the services selected such as data and header compresses, encryption, etc., and then builds the headers to deliver the packet.
3. IBM Mobility Client sends the packets back to the IP stack.
4. The IP stack forwards it to the network interface adapter.
5. The adapter sends the packets to the network addressed to IBM WebSphere Everyplace Connection Manager server.
6. IBM WebSphere Connection Manager receives the packets, decrypts, decompresses, and so on:
   a. It sends the packets to its destination intranet application server using the virtual IP as source.
   b. Alternatively, through a firewall, it sends packets to its destination Internet application server using the virtual IP as source. In this step the firewall adds a reference number to the packet to identify the response of the application server.

7. The application server responds and sends the response to the IBM Mobility Client:
   a. In this case, the network is prepared to deliver these packets to Connection Manager.
   b. Alternatively, when the application server responds and sends the response back to the firewall, the firewall forwards the response to the IBM Mobility Client, but the network is prepared to deliver these packets to Connection Manager.
8. Connection Manager encrypts, compresses, etc., and sends the packets back to the device network adapter.
9. The device network adapter sends the data to the IP stack.
10. The IP stack sends it to the IBM Mobility Client.
11. The Mobility Client decompress, decrypts, etc. the packet. It then delivers the packet to the IP stack.
12. The IP stack sends the packets to the application in the Mobility Client.

11.4 Packet headers

In each phase of the communication process, the headers will be changed to make available all the features provided by IBM WebSphere Everyplace Connection Manager. Figure 11-8 illustrates the structure of a TCP/IP standard packet.

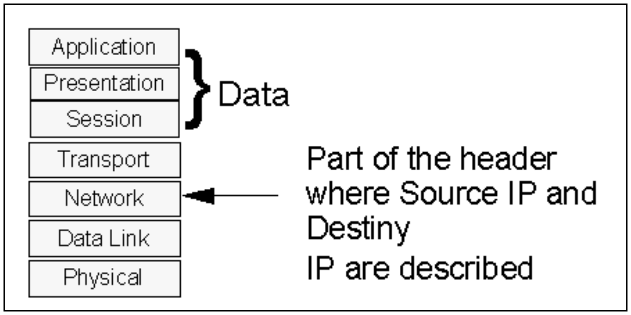


Figure 11-8 Packet headers - OSI Model

The three initial layers are the data itself and the other layers are respectively: *transport layer*, which inserts the TCP header; the *network layer,* which inserts the IP header; the *data link layer*, which encapsulates the packet in frames recognized by the network and adapter; and then the *physical layer*, which converts the packets into bits. The IBM Mobility Client works in layers 4 and 5, the transport and network layers respectively.

Before we start the explanation about the headers in an IBM WebSphere Everyplace Connection Manager environment, it is necessary to understand the basic knowledge about the headers in a simple and common network communication. The header's changes are illustrated in Figure 11-9.

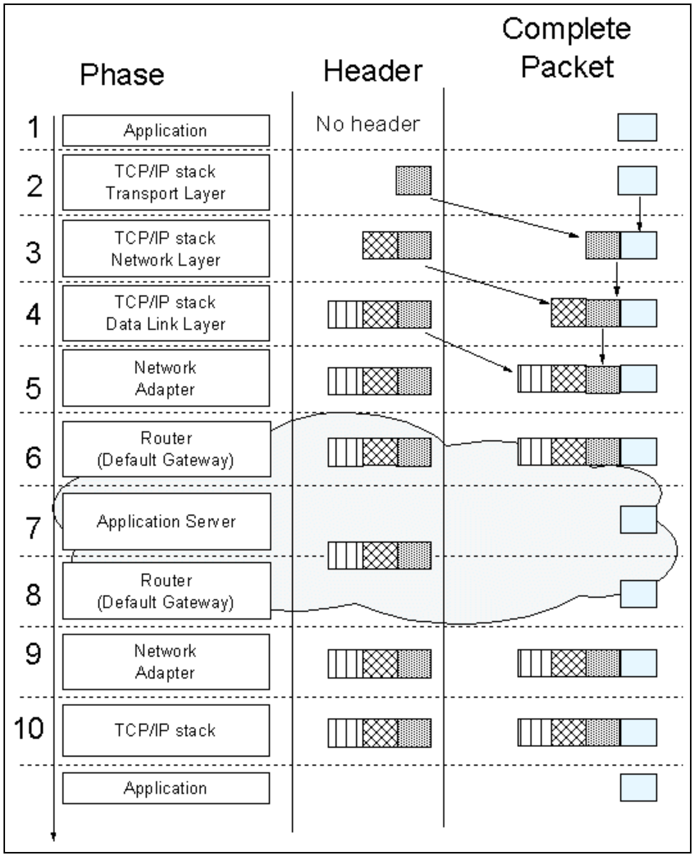


Figure 11-9 Header in simple network communication

The header's updates for a simple network communication happens as follows (the header's explanation is limited to relevant information only):

1. The application requests network communication for the TCP/IP stack by sending it some data.
2. The TCP/IP stack in its transport layer fragment the data and encapsulate it inserting the TCP header that includes the sequence in each of the fragments to make it possible to rebuild the data.

3. The TCP/IP stack in its network layer, checks the IP routes (as in Example 11-1 on page 194) to see what drivers must be used to build the IP header. The TCP/IP stack builds the header with the adapter information as the source IP and the application server as the destination IP.
4. The TCP/IP stack in its data link layer converts the packets in the adapter standard, and inserts the header with this information.
5. After the packet conversion into bits in the physical layer, the network adapter sends this packet to the network. If the destination IP is in the same subnet, the packet will be delivered directly to the application server adapter; else, if the destination IP is in another subnet, the packet will be sent for the default gateway that forwarded it to the default gateway of the remote subnet even using more routers in the path.
6. The packet is forwarded to the application server.
7. The application server removes the source header and interprets the data content. With this interpretation the application server selects the data that answers to this request and add its own packets, and builds its own header. This header is delivered back to the requester using as source IP the application server IP, and as destination IP using the IP of the user device network adapter.
8. The packet goes by the Internet using the same concept explained in item 5, and then is delivered to the user's device network adapter.
9. The network adapter delivers it to the TCP/IP stack.
10. - The TCP/IP stack removes the headers and deliver the packet to the application.

This flow explained is related to a usual communication without the IBM WebSphere Everyplace Connection Manager participation. The next header's flow is explained by using the IBM WebSphere Everyplace Connection Manager. The flow now is more complex and extensive. This explanation is related only to the packets that flow by the Connection Manager, and is shown in Figure 11-10.

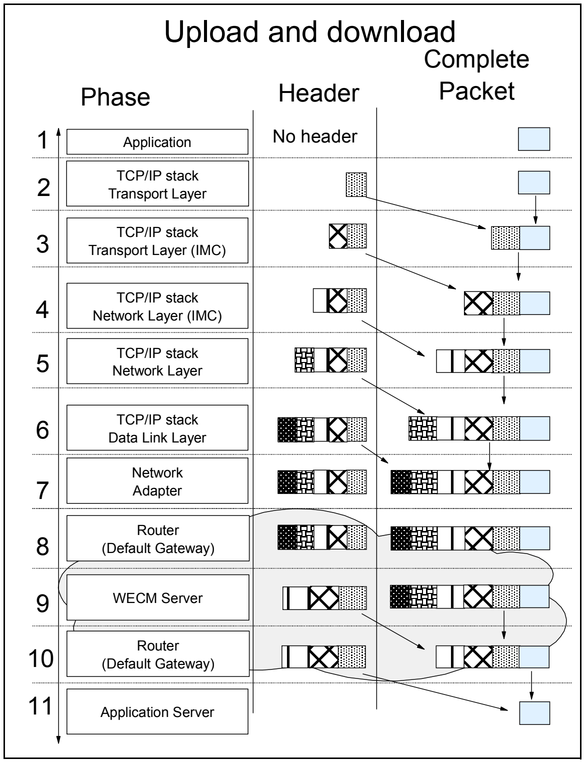


Figure 11-10 Headers in a Connection Manager environment

These are the steps:

1. The application requests network communication through the TCP/IP stack.
2. The TCP/IP stack adds the TCP header, and then uses the route to send packets to IBM Mobility Client.
3. IBM Mobility Client receives the packet and converts all the packets (data and the TCP header) into data. This data, depending on the services enabled, can be compressed and encrypted using different types of algorithms. For more information, refer to Chapter 10, “Mobility Clients” on page 171. Then, the IBM Mobility Client inserts its own transport layer using the TCP/IP stack to build a header with a UDP protocol.
4. The IBM Mobility Client works with the TCP/IP stack to build the network layer, inserting its own IP (virtual IP) as the source, and the application server IP as the destination. In this step the header compression occurs.

5. IBM Mobility Client works with the TCP/IP stack again to build a second network layer in which the source IP is the network adapter IP (Real IP) and the destination is the IBM WebSphere Everyplace Connection Manager IP.
6. The TCP/IP stack builds a data link layer to deliver data to the device's network adapter.
7. The network adapter sends the packet to the network.
8. The default gateway forwards the packet to IBM WebSphere Everyplace Connection Manager.
9. IBM WebSphere Everyplace Connection Manager receives the data and unpacks the UDP (step 3) header and the and the IP (Real IP) header. In this step, the data is decompressed and decrypted if it was selected on the client options. The Connection Manager forwards the packet with the transport layer (step 2) header and network layer header (step 4) to the application using its default gateway to reach the destination. In this step, the data may need to pass through a firewall, and this resource must be configured to accept the flow.
10. The network routers send the packet to the application server.
11. The application server receives the packet and removes the headers using its own TCP/IP stack.

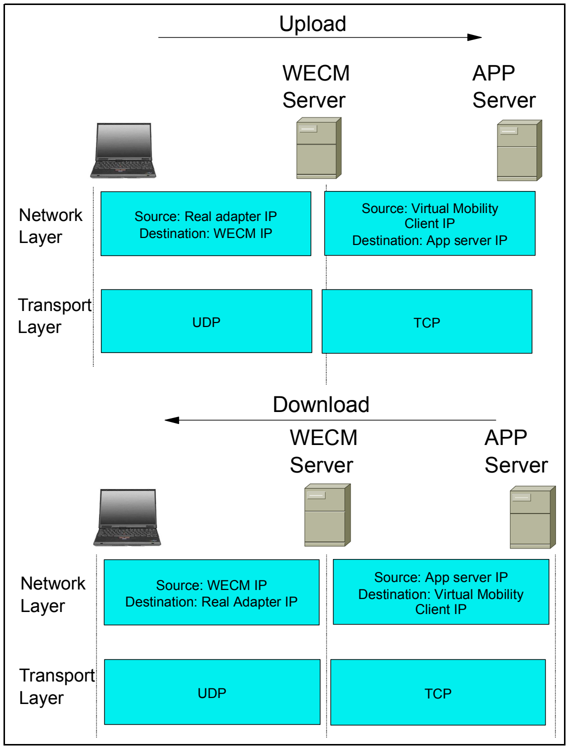


Figure 11-11 TCP and network headers

12

Mobility Client security

This chapter describes the security features in WebSphere Everyplace Connection Manager (WECM) associated with the Mobility Client connections.

The following topics implemented in WebSphere Everyplace Connection Manager (WECM) are discussed in this chapter:

- Authentication
- Certificates
- Symmetric key exchange
- Encryption
- WTLS for WLP clients

12.1 Overview

Connection Manager provides a secure wireless optimized IP tunnel; for more details see Chapter 9, "Mobile Access Services" on page 137. It uses the Wireless Layer Protocol (WLP) and optionally the Wireless Transport Layer Security (WTLS) protocol (Sun Solaris and Linux only) for an encrypted channel between a client device running the Mobility Client code and Connection Manager server.

Note: Other connections associated with other software components such as LDAP, Radius, and Gatekeeper can also be secured.

The following diagram gives an overview of the secure channels available in a secured Mobility Client architecture.

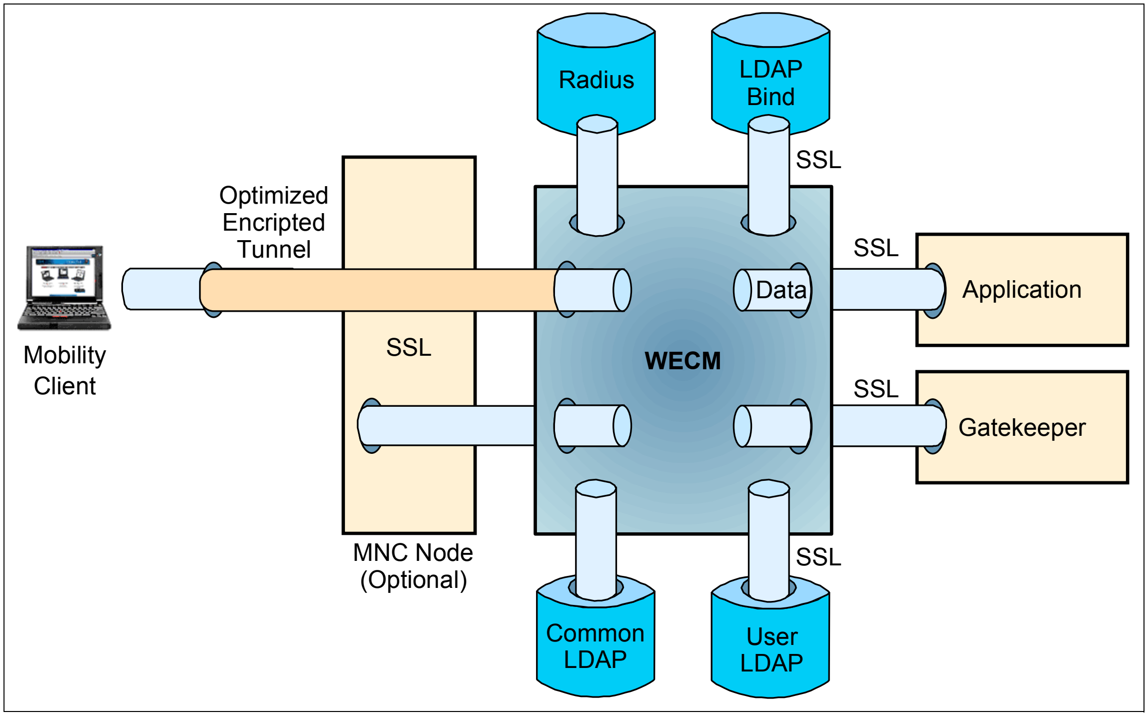


Figure 12-1 Mobility Client security

The Connection Manager product provides different ways to authenticate the client user and eventually the server when needed. In order to better understand a secure and optimized tunnel configuration we first illustrate all the Connection Manager resources and components involved and their connections. Figure 12-2 shows the framework between the Connection Manager resources in a client to server connection.

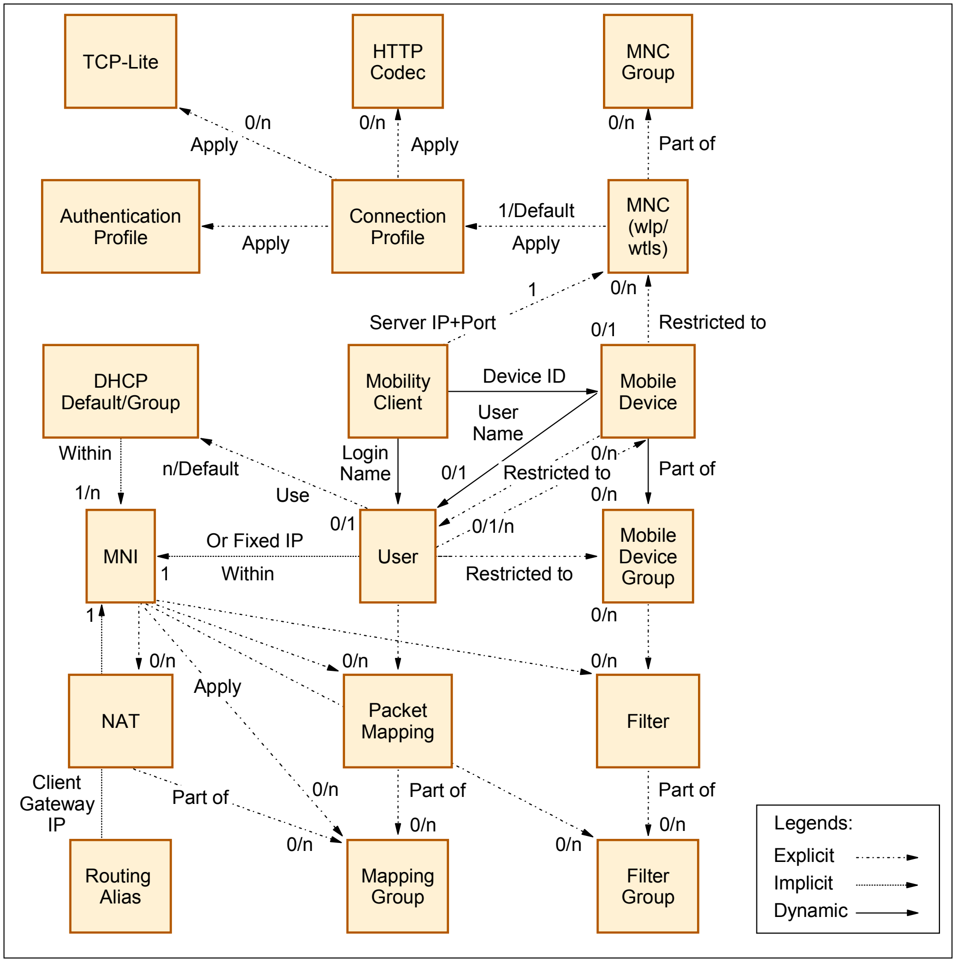


Figure 12-2 Mobility Client connection components

Note: Network Address Translation (NAT) is actually Port Address Translation (PAT).

12.2 Key exchange and authentication

Authentication is the process by which a connection endpoint may identify the other connection endpoint. For example, in a client to server connection the server may want to authenticate the client and vice versa.

Examples of client to server connections are:

- Web browser and a proxy server
- A proxy server and a Web server
- A Connection Manager VPN connection
- A SSL connection
- Others

The most common and basic scenario is when the Connection Manager server wants to authenticate Mobility Clients attempting to connect. In turn, and as a complement of this scenario, the Mobility Clients may also want to authenticate the server they are connecting to.

For a specific Mobile Network Connection (MNC), you will need to select a connection profile. No selection implies the use of the default connection profile. A security profile is part of the connection profile and it is made of a combination of different features. However, you should be aware that not all possible combinations are available.

The diagram shown in Figure 12-3 may help you to follow the connection setup process.

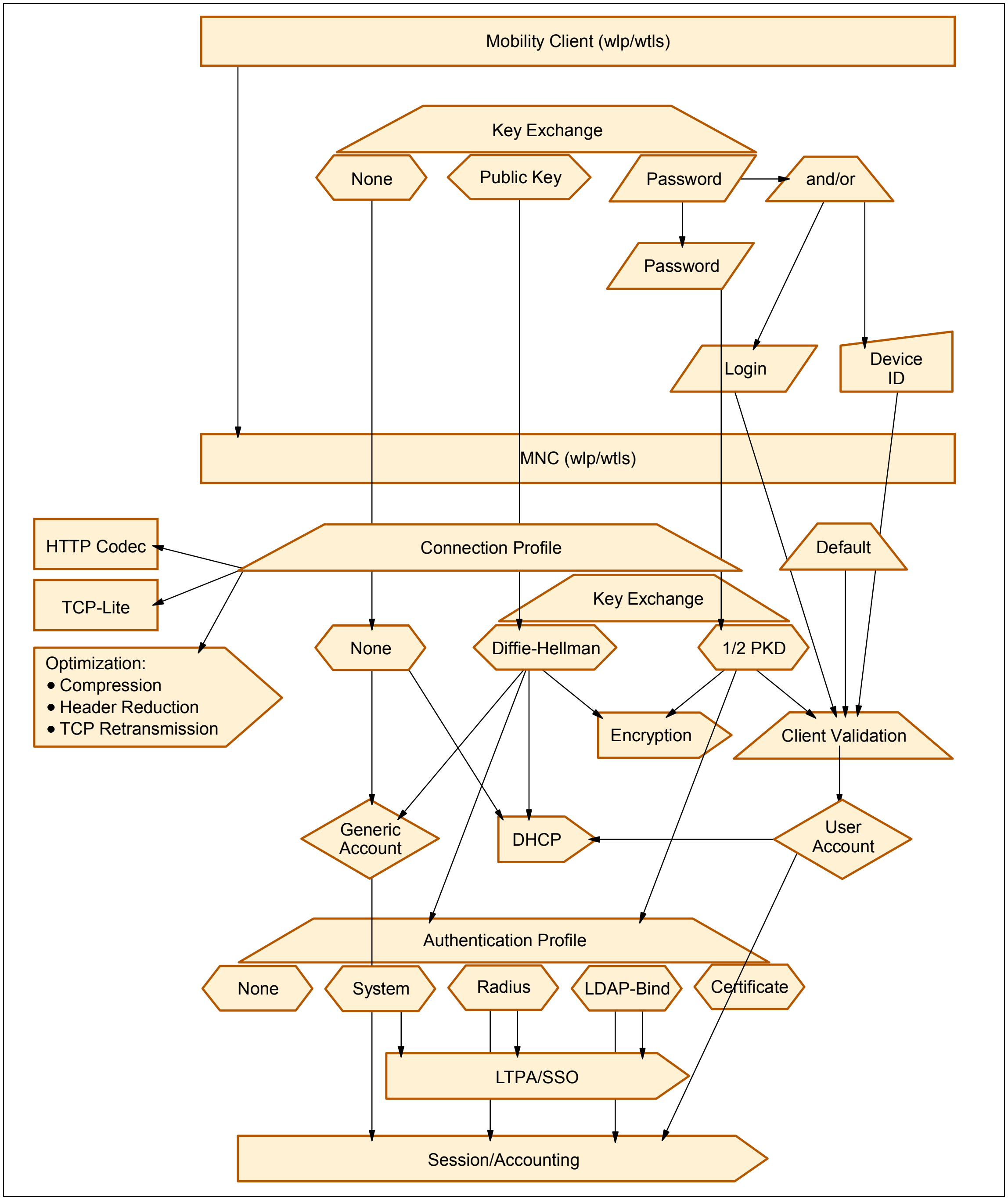


Figure 12-3 Connection setup process

Key exchange

The setup procedure to create an encrypted tunnel requires you to have an identical symmetric key at the client and server side of the connection. For Mobility Clients, this confidential exchange can be accomplished by using the *Diffie-Hellman protocol* or the *Key Distribution protocol*. In addition, the Key Distribution protocol also allows for some type of authentication.

Note: Diffie-Hellman allows for key exchange only, it does not do authentication. Key exchange is required for symmetric encryption.

The Single Party Key Distribution protocol (1PKD) allows for Mobility Client authentication. However, when the Two Party Key Distribution protocol (2PKDP) is used, it adds authentication of the Connection Manager server as well.

As a consequence, the output of the Key Distribution protocol is a user name which is used to name the session. The Diffie-Hellman protocol and the no key exchange option receives an anonymous session; in this case a temporary user named generic-x is created in the database for session purpose and in LDAP.

The Key Distribution protocol uses the password (saved in the Connection Manager LDAP server or in a specific user LDAP server) as a shared secret. Combined with random numbers and a hashing process, it provides an authentication token which is compared to be equal on both endpoints of the connection. That is, the password itself is never sent on the network. This token is the basis for the connection session key.

The Diffie-Hellman protocol creates public-private keys pairs at both endpoints and combines the public key from one endpoint with the private from the other endpoint to build the session key for the connection.

The session key is used as the basis for the tunnel encryption symmetric key. In addition, the session key may be reset periodically for improved security.

Client validation model

The Key Distribution protocol uses a validation model to associate a user name to a client connection. The default procedure is to scan the user LDAP repository using the Mobility Client login name as the search key.

The Gatekeeper allows you to define mobile device identifiers. It may be the source IP address (real one, not future VPN one) of the connection or other connection specific data. You may configure the validation model to get the mobile device identifier and to verify if it is an already known identifier.

You may also list in a user configuration, which mobile devices can be used to connect. Each mobile device entry may list the users names who are expected to connect from this device. When there is one and only one user name defined, you may configure the validation model to retrieve the connection user name from the connection Mobile Device identifier. Therefore, the user does not need to enter its login name on the Mobility Client prompt.

If the Mobility Client device supports it, you may also add a hardware device number (for example the Pocket PC internal number, Intel CPU Universal Unique ID) to the user configuration. This checking is not part of the validation model.

Secondary authentication

The key distribution protocol allows you to authenticate clients. The Diffie-Hellman protocol does not provide this option. In addition to these two key exchange protocols, you may also add another level of authorization, a failure to do this checking will also stop the connection setup.

You will have to define an authentication profile and configure the connection profile. The authentication profile may take advantage of the following resources:

- Radius server. Connection Manager server acts as a radius client.
- LDAP server. Connection Manager does a LDAP bind, no password request as with user LDAP server, but just a login challenge.
- Certificate database. Connection Manager may verify certificate validity period, certificate issuer or subject name (match with a user LDAP server entry).

Note: Radius and LDAP bind secondary authentication are also used by HTTP Access Services and WAP gateway services.

The authentication profile adds also to the possibility to create a Lightweight Third Party Authentication (LTPA) token and the option to enable Single Sign-On (SSO), except for the certificate authentication case.

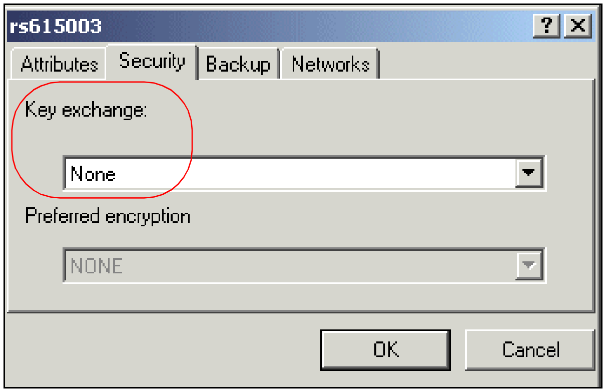
You may benefit from SSO without external authentication by using the system authentication profile. The default connection profile is set for the two party key distribution protocol for key exchange, the user validation model, and no secondary authentication profile.

12.3 Sample scenario

In this section some of the most typical scenarios are illustrated.

12.3.1 Anonymous

There is no authentication in this scenario, and there is no login or password. As a consequence, there may not be encryption either. The Mobility Client is configured for this anonymous connection as shown in Figure 12-4.

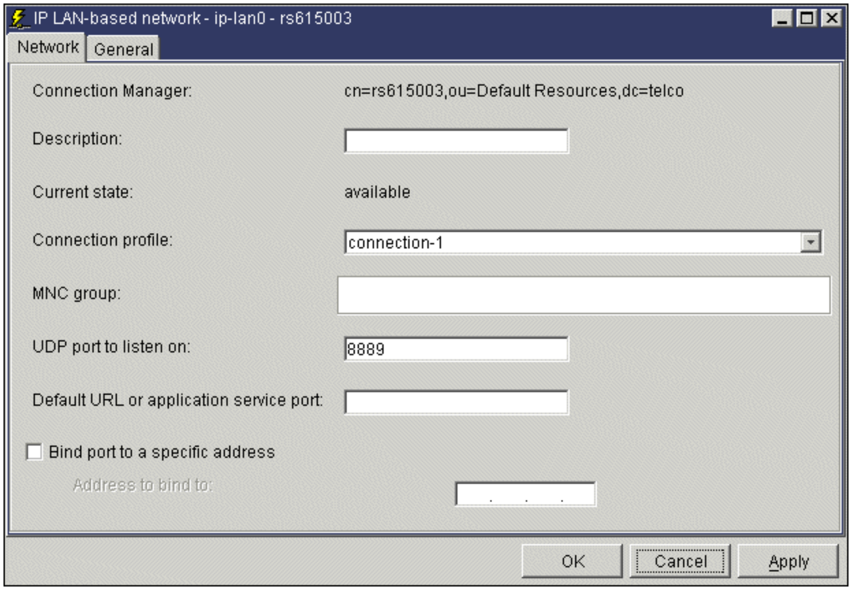


Key exchange: None

Preferred encryption: NONE

Figure 12-4 Key exchange option - None

The Connection Manager server allows you to configure for no authenticated connections on a specified IP address and port by configuring this option in the Mobile Network Connection (MNC).



Connection Manager: cn=rs615003,ou=Default Resources,dc=telco

Description:

Current state: available

Connection profile: connection-1

MNC group:

UDP port to listen on: 8889

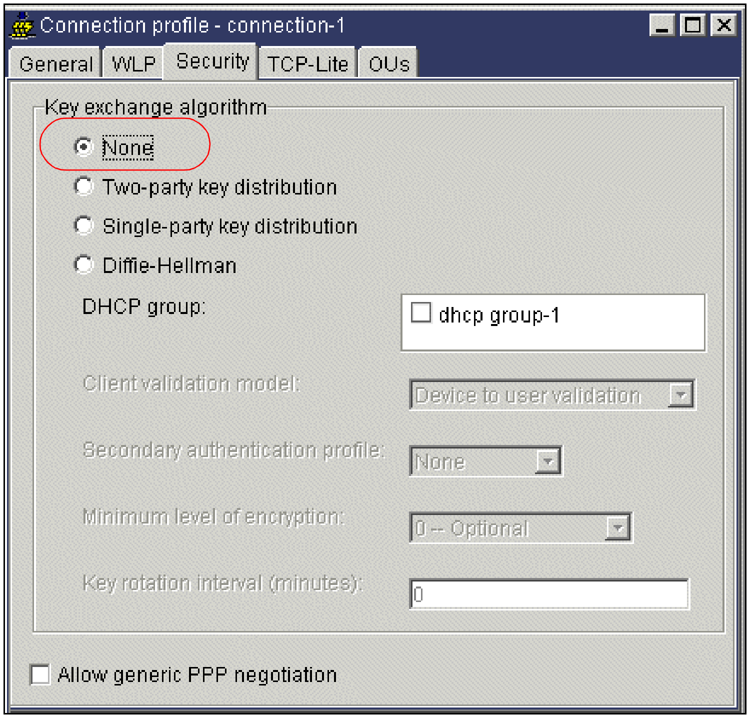
Default URL or application service port:

□ Bind port to a specific address

Address to bind to: . . .

Figure 12-5 Mobile Network Connection (MNC)

In this example, you are not using the default Connection profile, but a profile with no security as shown in Figure 12-6.



Connection profile - connection-1

Key exchange algorithm

- ● None
- ○ Two-party key distribution
- ○ Single-party key distribution
- ○ Diffie-Hellman

DHCP group: ☐ dhcp group-1

Client validation model: Device to user validation

Secondary authentication profile: None

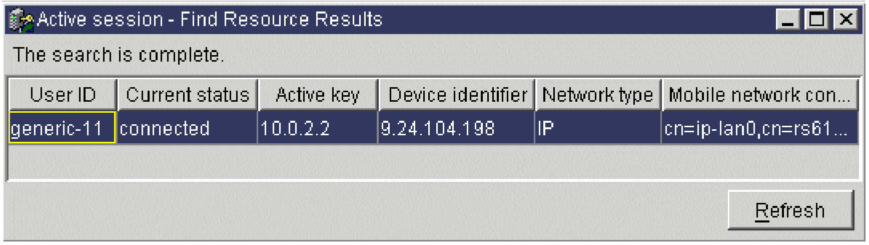
Minimum level of encryption: 0 -- Optional

Key rotation interval (minutes): 0

☐ Allow generic PPP negotiation

Figure 12-6 Key exchange algorithm - None

A entry is added to the session table with the name generic-xxx. A temporary user entry with the same name is also created in the common LDAP server. See also “As a result, the server creates incremental temporary user names called generic-xxx in the session table. For example, Figure 12-21 shows a generic user session for user generic-13.” on page 229 within Diffie-Hellman key exchange.



Active session - Find Resource Results

The search is complete.

User ID	Current status	Active key	Device identifier	Network type	Mobile network con...
generic-11	connected	10.0.2.2	9.24.104.198	IP	cn=ip-lan0,cn=rs61...

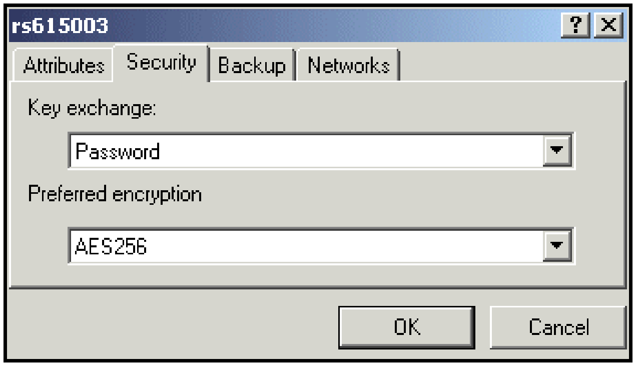
Figure 12-7 Generic user session

Note: In this sample scenario you obtain an IP address and you may still enable data compression.

12.3.2 Basic

This is the default scenario. You receive all the security features available. The Mobility Client and the Connection Manager server are authenticated to each other. The encryption protocols available are dependent of the client hardware.

The client configuration should match the server MNC configuration.



Key exchange:

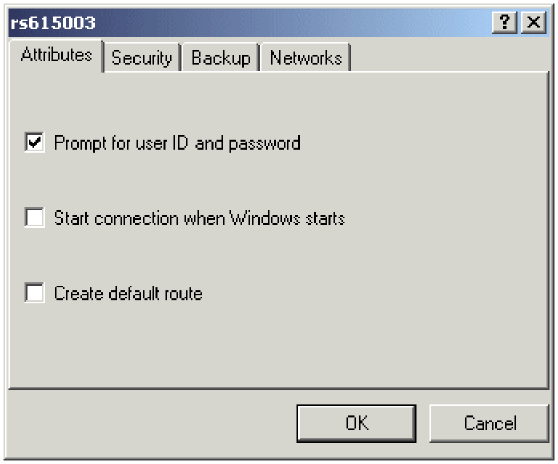
Password

Preferred encryption

AES256

Figure 12-8 Client password for key exchange

The client may request the login and password pop-up window the first time to fill in the fields.



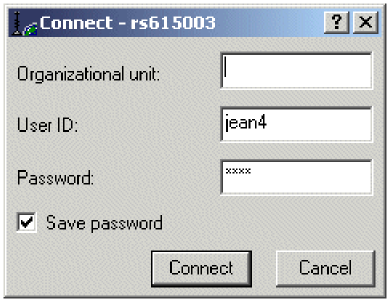
☑ Prompt for user ID and password

□ Start connection when Windows starts

□ Create default route

Figure 12-9 Client pop-up requested

You will then provide the user ID and password and optionally check the **Save password** option so you no longer are requested to enter it again. See Figure 12-10.



Connect - rs615003

Organizational unit:

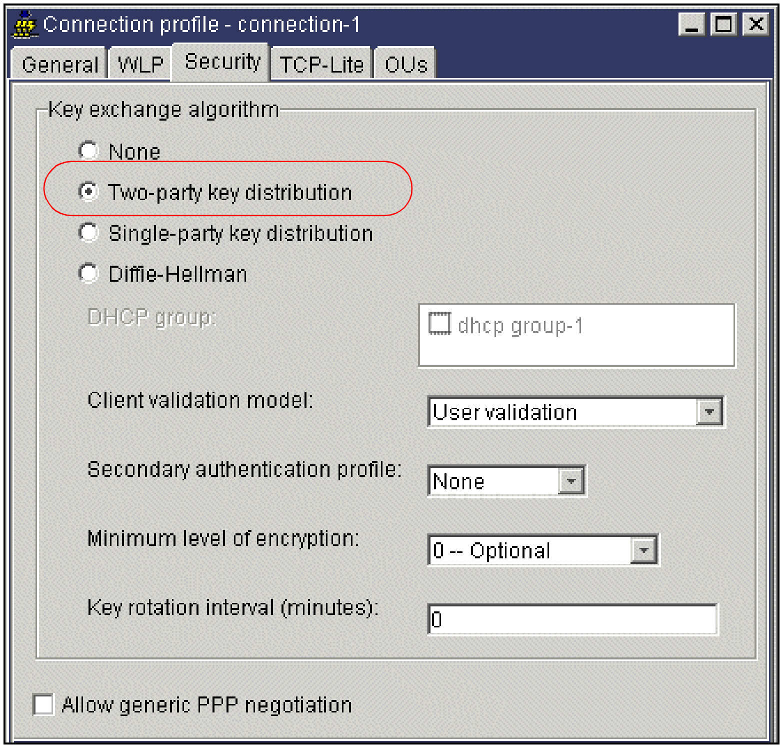
User ID: jean4

Password: ****

☑ Save password

Figure 12-10 Client login and password pop-up window

This connection profile maps the default profile.



Connection profile - connection-1

Key exchange algorithm

○ None

● Two-party key distribution

○ Single-party key distribution

○ Diffie-Hellman

DHCP group: □ dhcp group-1

Client validation model: User validation

Secondary authentication profile: None

Minimum level of encryption: 0 -- Optional

Key rotation interval (minutes): 0

□ Allow generic PPP negotiation

Figure 12-11 Server two-party key distribution exchange

12.3.3 Device identifier

This scenario uses the Connection Manager capability to recover the user name from the client device. During the login process, the server tries to match a device identifier with a list of predefined entries. In addition, a few extra checks are made.

For example, a device may be restricted to be used by some specific user. Or, a user may be restricted to use some devices. When a device is restricted to be used by one and only one user, it allows you to retrieve its user name from one of its devices.

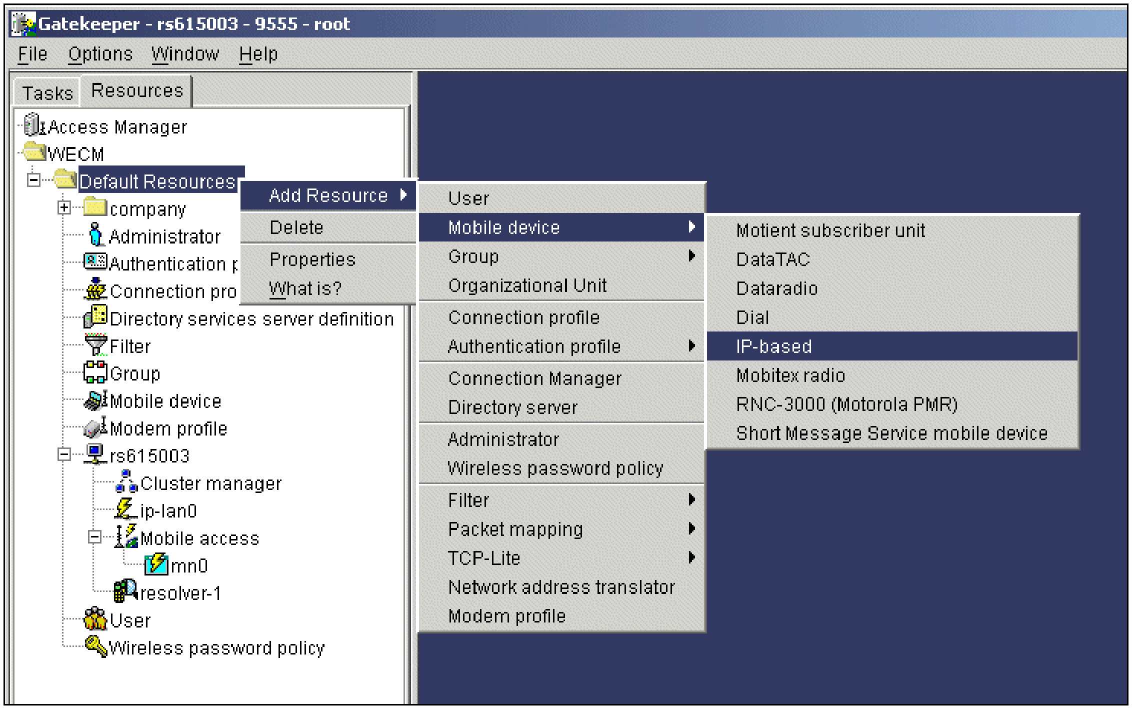
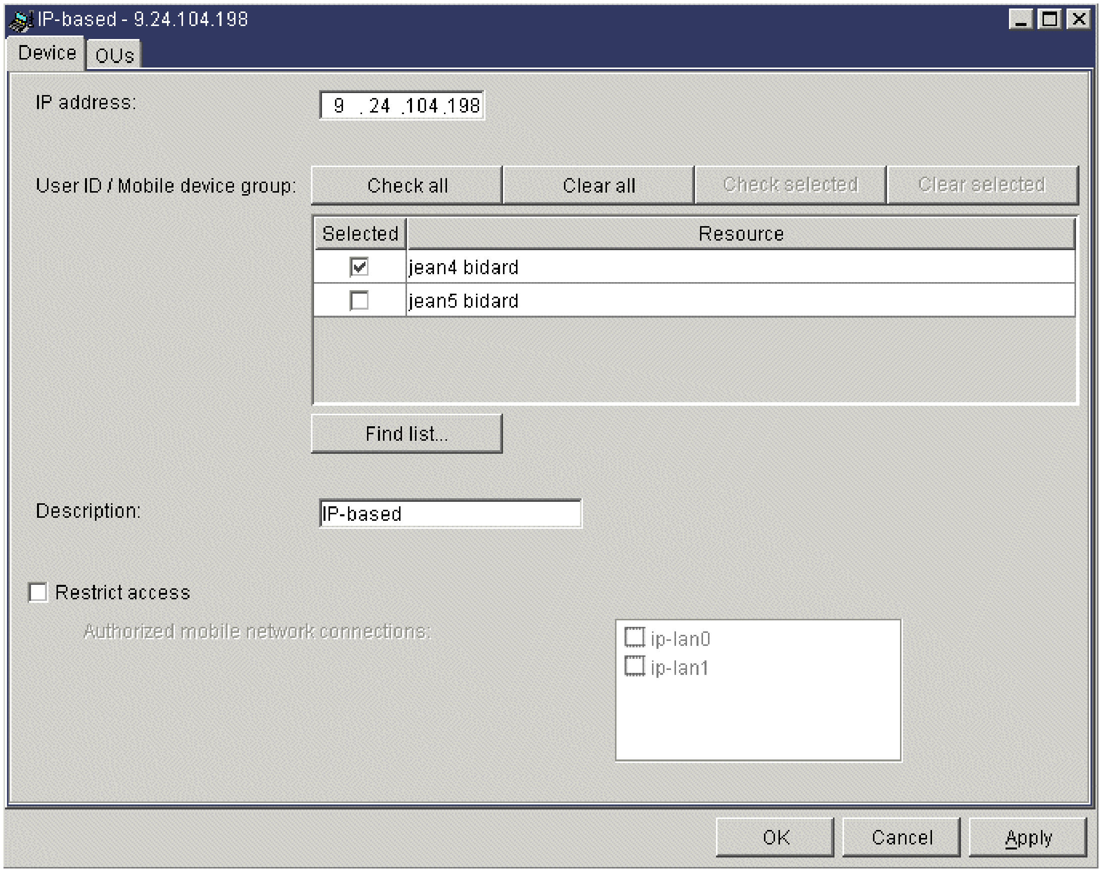


Figure 12-12 Device identifier setup

For example, Figure 12-13 illustrates the use of the source IP address of the Mobility Client as seen by the server to associate only one user to this device.



IP address: 9 .24 .104 .198

User ID / Mobile device group:

Selected	Resource
☑	jean4 bidard
☐	jean5 bidard

Description: IP-based

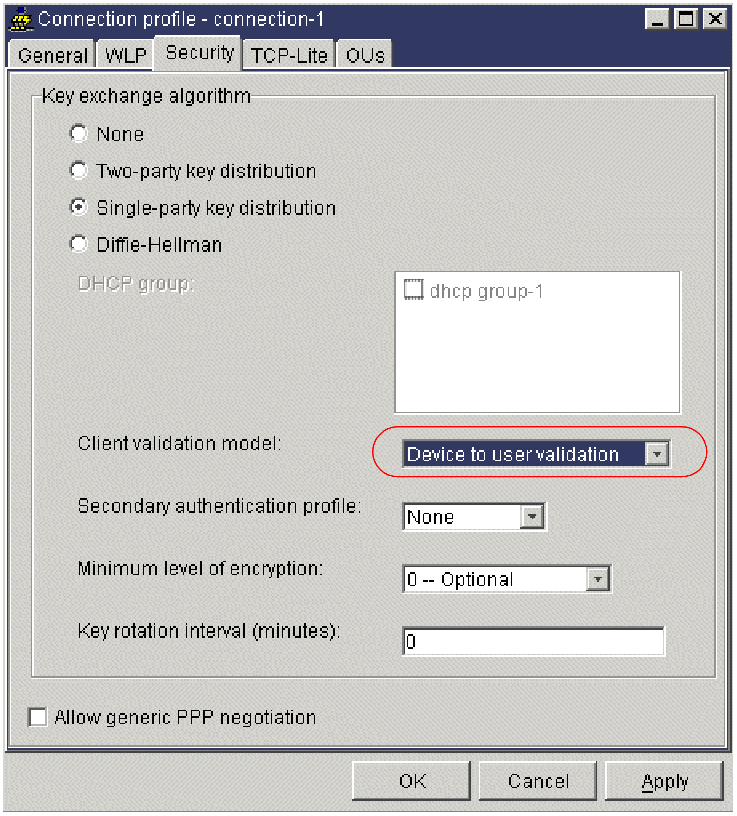
☐ Restrict access

Authorized mobile network connections:

☐ ip-lan0
☐ ip-lan1

Figure 12-13 IP based device identifier

Figure 12-4 shows the configuration for a *Device to user validation* model.



Connection profile - connection-1

Key exchange algorithm

○ None
○ Two-party key distribution
● Single-party key distribution
○ Diffie-Hellman

DHCP group: □ dhcp group-1

Client validation model: Device to user validation

Secondary authentication profile: None

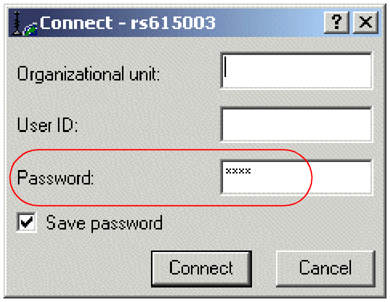
Minimum level of encryption: 0 -- Optional

Key rotation interval (minutes): 0

□ Allow generic PPP negotiation

Figure 12-14 Device to user validation

In this scenario, the client no longer needs to enter its login name to be authenticated, and only the password is entered as illustrated in Figure 12-15.



Connect - rs615003

Organizational unit:

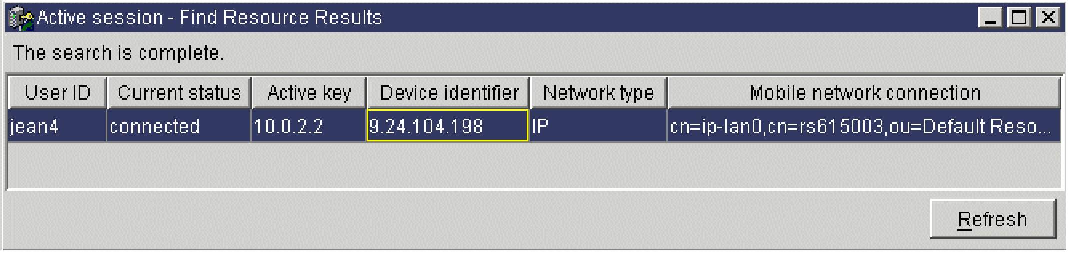
User ID:

Password: ****

☑ Save password

Figure 12-15 Pop-up window requesting a password only

Figure 12-16 shows the session table indicating the retrieved user name. Notice that the session table displays the device identifier.



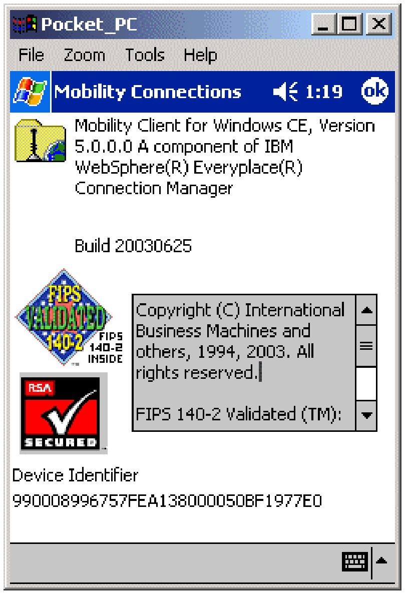
The search is complete.

User ID	Current status	Active key	Device identifier	Network type	Mobile network connection
jean4	connected	10.0.2.2	9.24.104.198	IP	cn=ip-lan0,cn=rs615003,ou=Default Reso...

Figure 12-16 Connected user session entry

12.3.4 Device unique number

If supported, the user may also define a hardware device unique number. It may be for example the Intel CPU Universal Unique ID, or a specific handheld hardware number. It may not be used for authentication, but just as additional checking. This is not a duplicate of the device identifier, which is a logical identifier. In addition, you may check its availability by looking in the Mobility Client Help and About options. For example, Figure 12-17 illustrates a PocketPC device unique number.



Mobility Client for Windows CE, Version 5.0.0.0 A component of IBM WebSphere(R) Everyplace(R) Connection Manager

Build 20030625

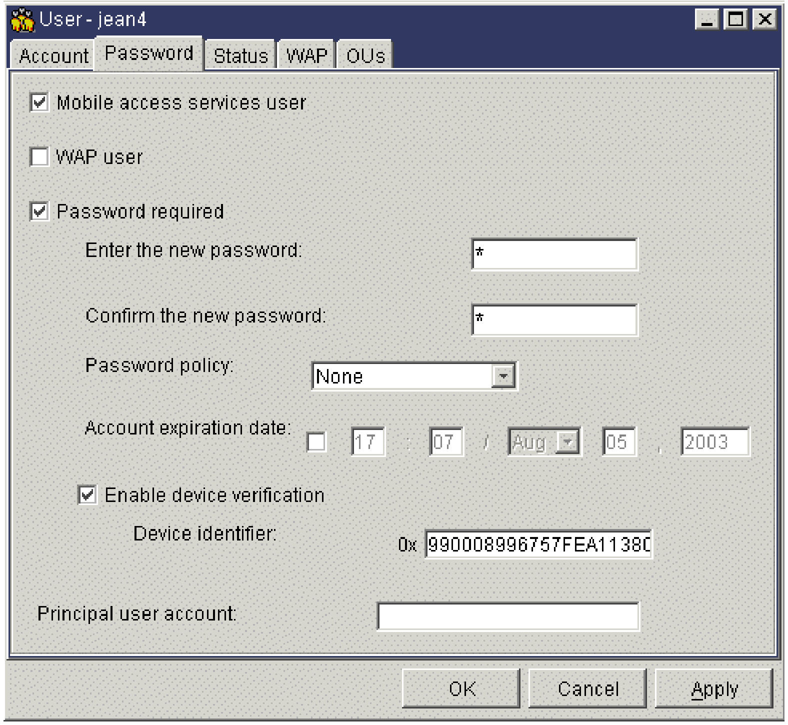
Copyright (C) International Business Machines and others, 1994, 2003. All rights reserved.

FIPS 140-2 Validated (TM):

Device Identifier
990008996757FEA138000050BF1977E0

Figure 12-17 Pocket PC device unique number

This hardware unique number may be pre-configured in the Gatekeeper user properties, or it is automatically entered during the connection setup.



☑ Mobile access services user

☐ WAP user

☑ Password required

Enter the new password: *

Confirm the new password: *

Password policy: None

Account expiration date: ☐ 17 : 07 / Aug 05 , 2003

☑ Enable device verification

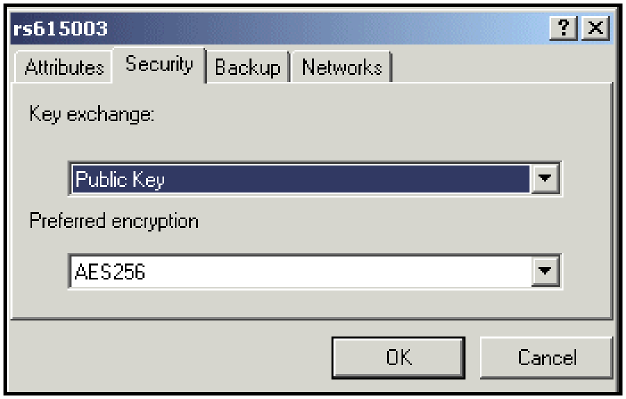
Device identifier: 0x 990008996757FEA1138C

Principal user account:

Figure 12-18 Unique number display

12.3.5 Diffie-Hellman

This protocol allows you to do an anonymous secured key exchange. When using this protocol the user is not authenticated, but data encryption can be used. The Mobility Client will need to be configured to use this key exchange protocol, for example see Figure 12-19.



Key exchange:

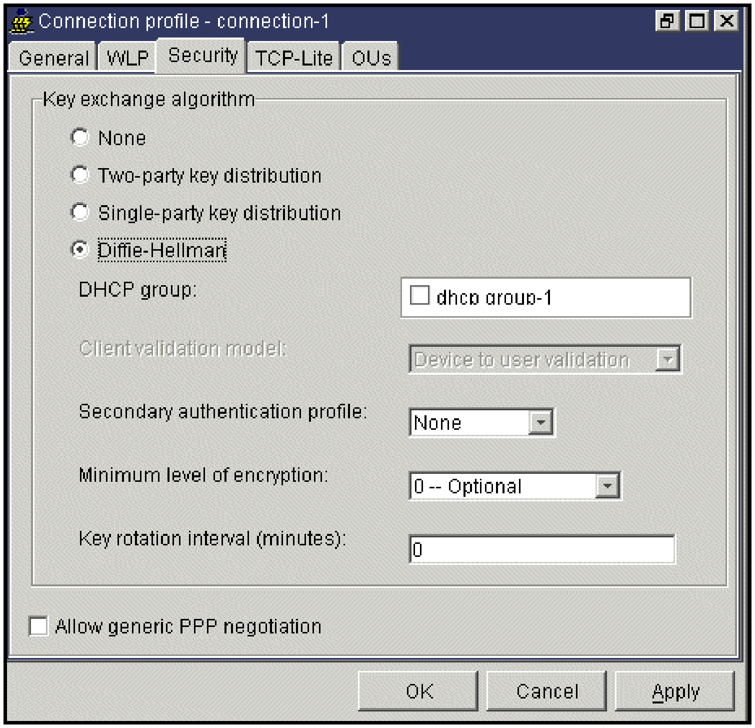
Public Key

Preferred encryption

AES256

Figure 12-19 Client Diffie-Hellman setup

In this scenario, the server cannot be configured to use a client validation model as shown in Figure 12-20.



Key exchange algorithm

○ None

○ Two-party key distribution

○ Single-party key distribution

● Diffie-Hellman

DHCP group: □ dhcp group-1

Client validation model: Device to user validation

Secondary authentication profile: None

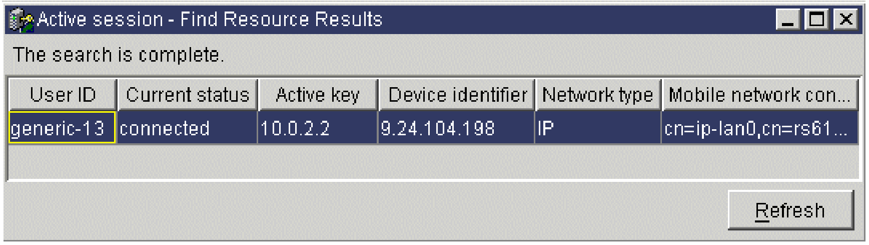
Minimum level of encryption: 0 -- Optional

Key rotation interval (minutes): 0

□ Allow generic PPP negotiation

Figure 12-20 Server Diffie-Hellman setup

As a result, the server creates incremental temporary user names called *generic-xxx* in the session table. For example, Figure 12-21 shows a generic user session for user generic-13.



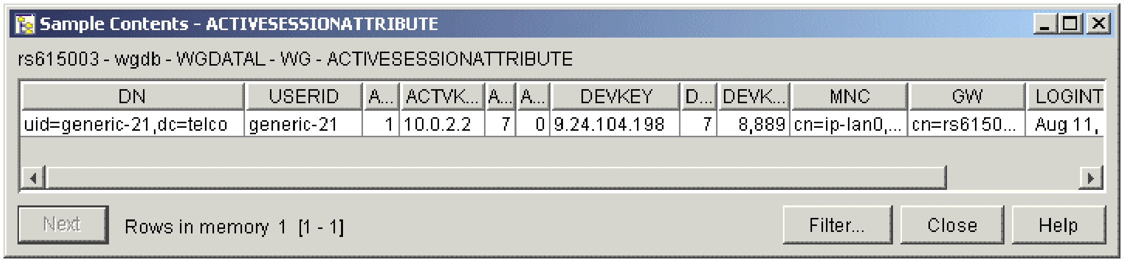
Active session - Find Resource Results

The search is complete.

User ID	Current status	Active key	Device identifier	Network type	Mobile network con...
generic-13	connected	10.0.2.2	9.24.104.198	IP	cn=ip-lan0,cn=rs61...

Figure 12-21 Generic user session

This database view shows that this user distinguished name is not a standard one.



Sample Contents - ACTIVESESSIONATTRIBUTE

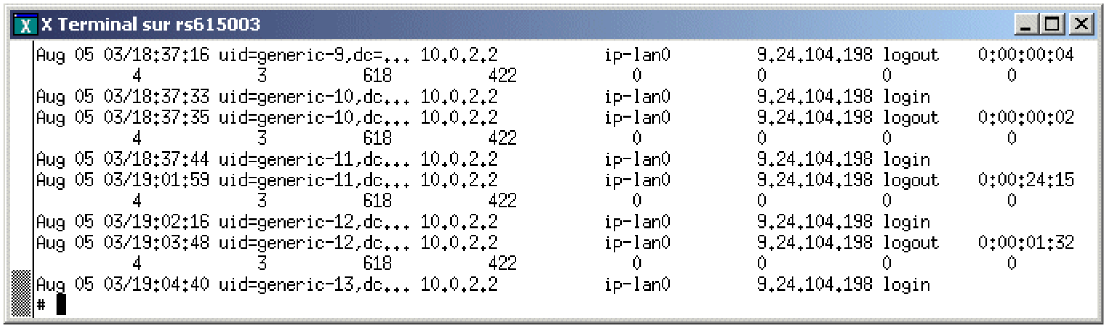
rs615003 - wgdb - WGDATAL - WG - ACTIVESESSIONATTRIBUTE

DN	USERID	A...	ACTVK...	A...	A...	DEVKEY	D...	DEVK...	MNC	GW	LOGINT
uid=generic-21,dc=telco	generic-21	1	10.0.2.2	7	0	9.24.104.198	7	8,889	cn=ip-lan0,...	cn=rs6150...	Aug 11,

Rows in memory 1 [1 - 1]

Figure 12-22 Database generic user entry (db2 Control Center)

It may be used for accounting purposes, filtered by MNC.



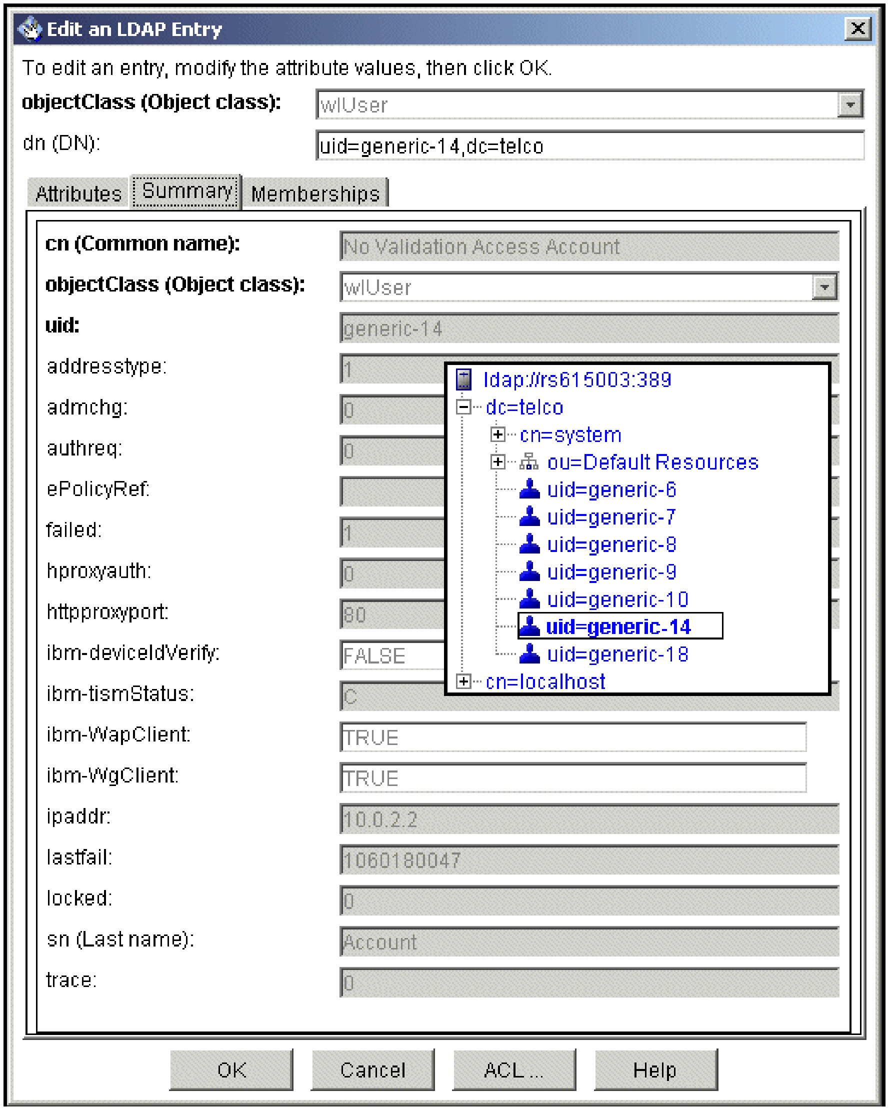
X Terminal sur rs615003

```
Aug 05 03/18:37:16 uid=generic-9,dc=... 10.0.2.2          ip-lan0          9.24.104.198 logout    0:00:00:04
           4           3          618         422          0               0           0              0
Aug 05 03/18:37:33 uid=generic-10,dc... 10.0.2.2          ip-lan0          9.24.104.198 login
Aug 05 03/18:37:35 uid=generic-10,dc... 10.0.2.2          ip-lan0          9.24.104.198 logout    0:00:00:02
           4           3          618         422          0               0           0              0
Aug 05 03/18:37:44 uid=generic-11,dc... 10.0.2.2          ip-lan0          9.24.104.198 login
Aug 05 03/19:01:59 uid=generic-11,dc... 10.0.2.2          ip-lan0          9.24.104.198 logout    0:00:24:15
           4           3          618         422          0               0           0              0
Aug 05 03/19:02:16 uid=generic-12,dc... 10.0.2.2          ip-lan0          9.24.104.198 login
Aug 05 03/19:03:48 uid=generic-12,dc... 10.0.2.2          ip-lan0          9.24.104.198 logout    0:00:01:32
           4           3          618         422          0               0           0              0
Aug 05 03/19:04:40 uid=generic-13,dc... 10.0.2.2          ip-lan0          9.24.104.198 login
#
```

Figure 12-23 wg_acct output

It also creates a LDAP user entry in the common LDAP tree under the suffix distinguished name (not in the default resources branch).

The LDAP user entry is shown in Figure 12-24. This entry allows basic actions on remotely defined users such as the lock feature.



Edit an LDAP Entry

To edit an entry, modify the attribute values, then click OK.

objectClass (Object class):	wlUser
dn (DN):	uid=generic-14,dc=telco

cn (Common name):	No Validation Access Account
objectClass (Object class):	wlUser
uid:	generic-14
addresstype:	1
admchg:	0
authreq:	0
ePolicyRef:	
failed:	1
hproxyauth:	0
httpproxyport:	80
ibm-deviceIdVerify:	FALSE
ibm-tismStatus:	C
ibm-WapClient:	TRUE
ibm-WgClient:	TRUE
ipaddr:	10.0.2.2
lastfail:	1060180047
locked:	0
sn (Last name):	Account
trace:	0

```
ldap://rs615003:389
dc=telco
    cn=system
    ou=Default Resources
    uid=generic-6
    uid=generic-7
    uid=generic-8
    uid=generic-9
    uid=generic-10
    uid=generic-14
    uid=generic-18
cn=localhost
```

Figure 12-24 Generic user LDAP entry

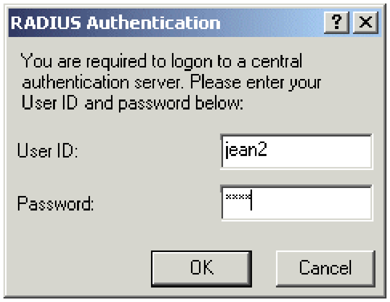
12.4 Additional authentication scenario

In addition to the first level of authentication, you may have a second level for authorization, it intends to use other components external to Connection Manager such as remote databases or root CA.

12.4.1 System

This predefined authentication profile does the same as the ldap-bind, but with the Connection Manager user LDAP server already configured in the Access Manager. It provides LTPA and SSO features.

After the optional user validation pop-up, you get the LDAP (named radius!) challenge pop-up (it may take some time to get it).



RADIUS Authentication

You are required to logon to a central authentication server. Please enter your User ID and password below:

User ID: jean2

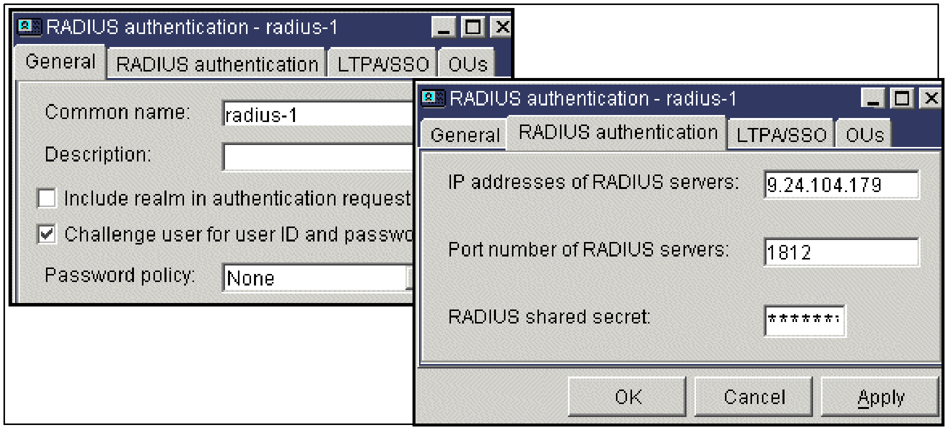
Password: ****

Figure 12-25 Mobility Client System challenge pop-up

12.4.2 Radius and secure ID

The Connection Manager server acts as a radius client and challenges a radius server. This radius server is not necessarily the one used for accounting, and is defined in the Connection Manager.

You may set the radius server to work with a secure ID token server, and you will need to create a radius authentication profile. The radius authentication configuration is shown in Figure 12-26.



Common name: radius-1

Description:

□ Include realm in authentication request

☑ Challenge user for user ID and passwo

Password policy: None

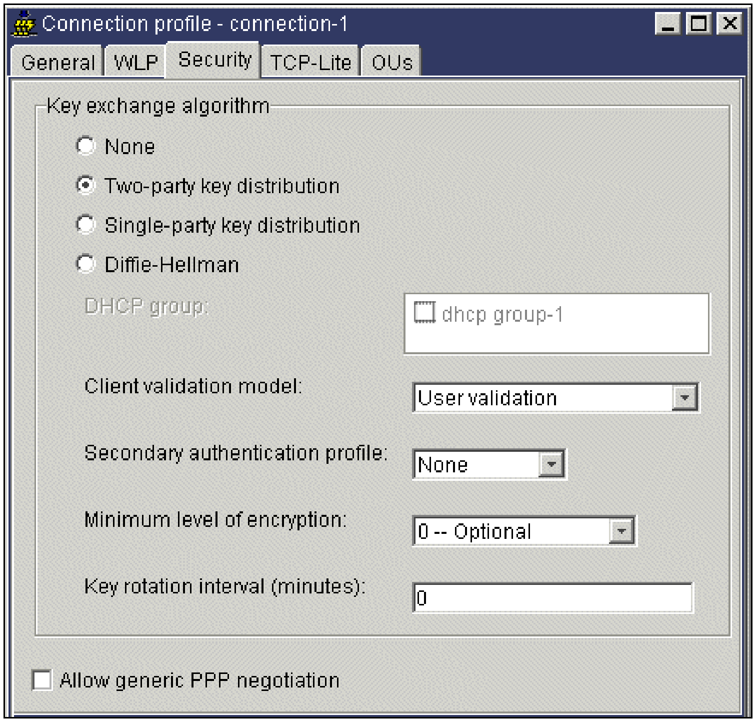
IP addresses of RADIUS servers: 9.24.104.179

Port number of RADIUS servers: 1812

RADIUS shared secret: *******

Figure 12-26 Radius authentication configuration

You may check out the challenge check box to use the same login name and password for the user validation model and the radius challenge.



Key exchange algorithm

○ None

● Two-party key distribution

○ Single-party key distribution

○ Diffie-Hellman

DHCP group: □ dhcp group-1

Client validation model: User validation

Secondary authentication profile: None

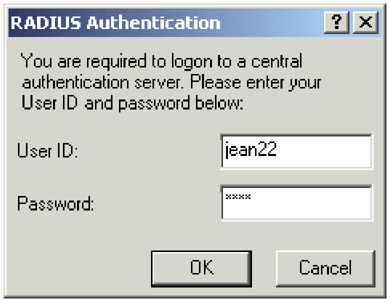
Minimum level of encryption: 0 -- Optional

Key rotation interval (minutes): 0

□ Allow generic PPP negotiation

Figure 12-27 Radius connection profile

After the optional user validation pop-up window, you will receive the optional radius challenge pop-up window shown in Figure 12-28.



RADIUS Authentication

You are required to logon to a central authentication server. Please enter your User ID and password below:

User ID: jean22

Password: ****

Figure 12-28 Mobility Client radius challenge pop-up window

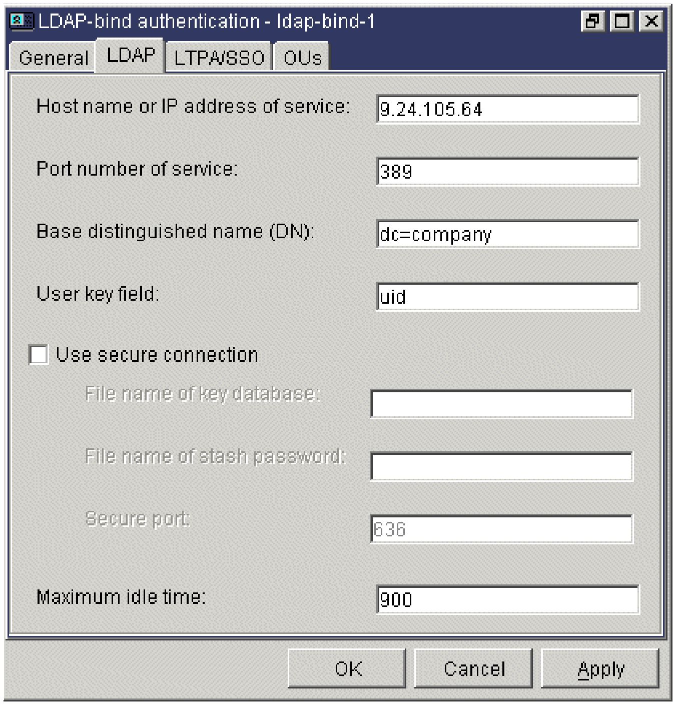
You may also want see “Session user” on page 236 for information about session tables.

12.4.3 LDAP bind

The Connection Manager server challenges an external LDAP server (not the user LDAP server) by running an authorization challenge. However, it does not try to get any LDAP attributes at this time.

For this process you will need to define the key in the remote LDAP server. For example, the attribute name: uid, cn, mail(=default). In addition, a secure connection can also be established using Secure Sockets Layer (SSL), this allows you to authenticate the LDAP server.

Figure 12-29 shows a sample LDAP-bind authentication profile.



Host name or IP address of service: 9.24.105.64

Port number of service: 389

Base distinguished name (DN): dc=company

User key field: uid

□ Use secure connection

File name of key database:

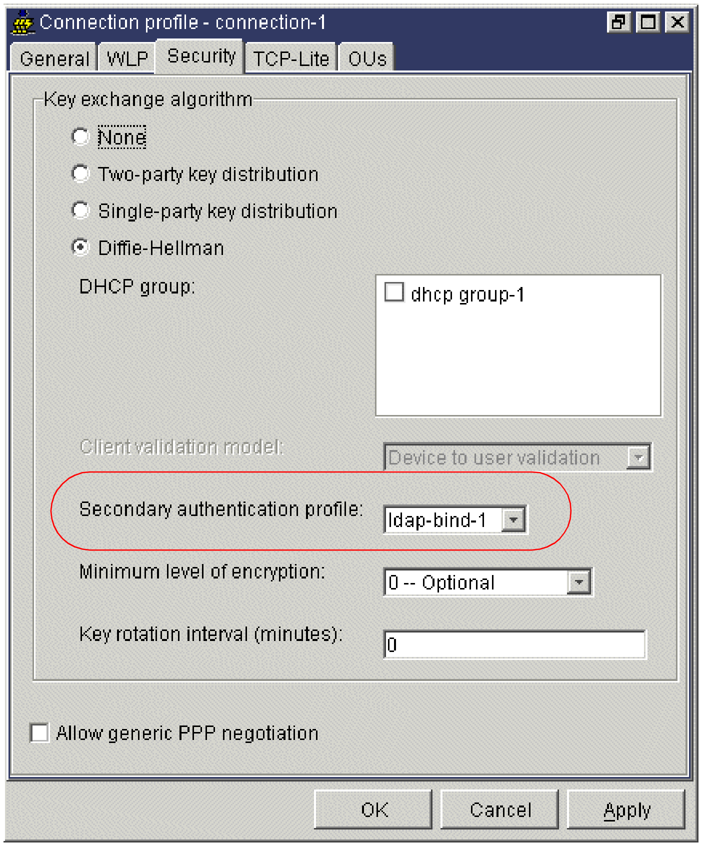
File name of stash password:

Secure port: 636

Maximum idle time: 900

Figure 12-29 Sample LDAP-bind authentication profile (ldap-bind-1)

The LDAP-bind authentication profile is selected in the connection profile as illustrated in Figure 12-30.



Connection profile - connection-1

Key exchange algorithm

○ None
○ Two-party key distribution
○ Single-party key distribution
● Diffie-Hellman

DHCP group: □ dhcp group-1

Client validation model: Device to user validation

Secondary authentication profile: ldap-bind-1

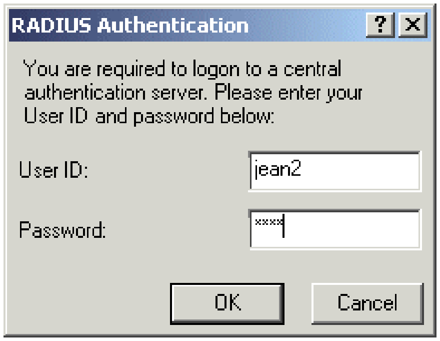
Minimum level of encryption: 0 -- Optional

Key rotation interval (minutes): 0

□ Allow generic PPP negotiation

Figure 12-30 LDAP-bind connection profile

After the optional user validation pop-up window, you will receive the LDAP (named RADIUS!) challenge pop-up (it may take you some time to see it).



RADIUS Authentication

You are required to logon to a central authentication server. Please enter your User ID and password below:

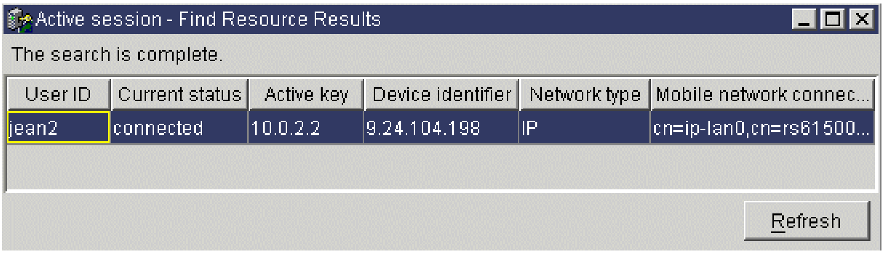
User ID: jean2

Password: ****

Figure 12-31 Mobility Client ldap-bind challenge pop-up window

Session user

As you get connected, Connection Manager creates an entry in the session table. When connected through the user validation model, it uses the same user name in the table entry. When connected with Diffie-Hellman, it uses the second level authentication login name. See Figure 12-32.



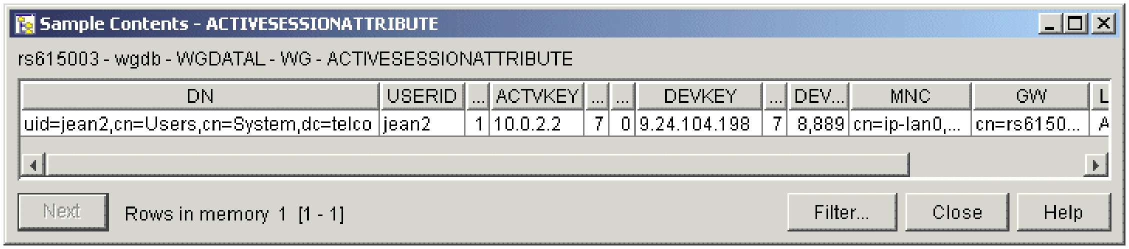
Active session - Find Resource Results

The search is complete.

User ID	Current status	Active key	Device identifier	Network type	Mobile network connec...
jean2	connected	10.0.2.2	9.24.104.198	IP	cn=ip-lan0,cn=rs61500...

Figure 12-32 User session

Notice also that this database view shows that the user Distinguished Name is not standard.



Sample Contents - ACTIVESESSIONATTRIBUTE

rs615003 - wgdb - WGDATAL - WG - ACTIVESESSIONATTRIBUTE

DN	USERID	...	ACTVKEY	...	...	DEVKEY	...	DEV...	MNC	GW	L
uid=jean2,cn=Users,cn=System,dc=telco	jean2	1	10.0.2.2	7	0	9.24.104.198	7	8,889	cn=ip-lan0,...	cn=rs6150...	A

Rows in memory 1 [1 - 1]

Figure 12-33 Database user entry

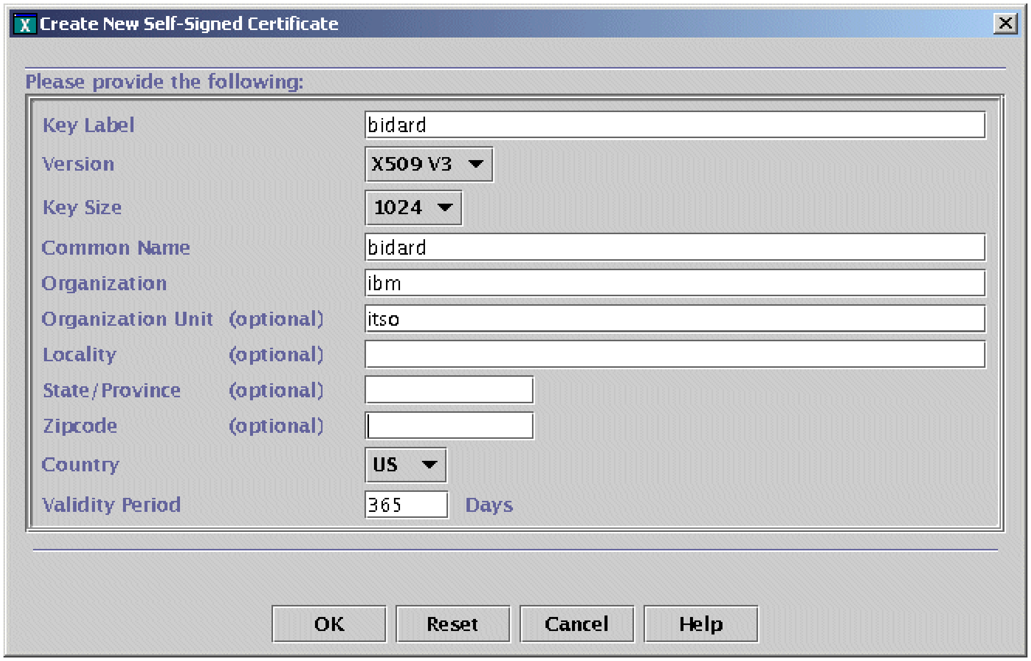
12.4.4 Certificate based

The Connection Manager server challenges a local certificate database with a certificate provided by the Mobility Client.

The server can check the following client certificate authenticity as follows:

- The certificate date is within a valid range.
- A Certificate Authority (CA) root certificate stored in a local key database certifies the client certificate.
- The client certificate subject name matches a user LDAP server entry.

The provided wg_ikeyman tool is used to manage the local key databases. You also use this tool to create a self-signed certificate. However, it is recommended that for production systems that you send a request to an official CA and get a root signed certificate for better security. For example, Figure 12-34 illustrates how wg_ikeyman is used to create a self-signed certificate.

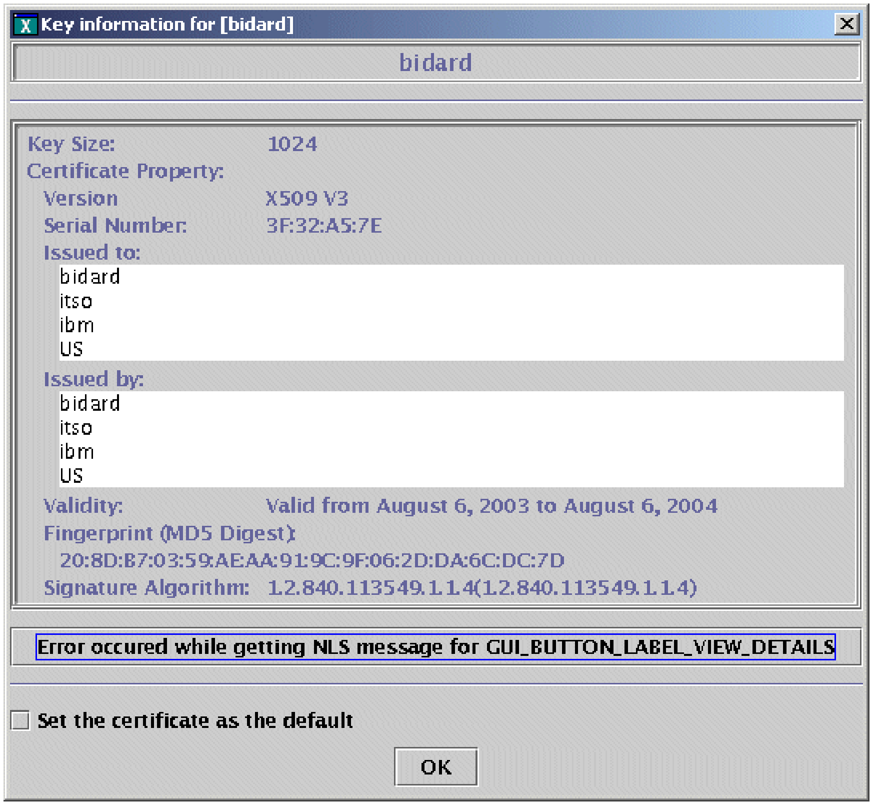

Create New Self-Signed Certificate

Please provide the following:

Key Label	bidard
Version	X509 V3
Key Size	1024
Common Name	bidard
Organization	ibm
Organization Unit (optional)	itso
Locality (optional)	
State/Province (optional)	
Zipcode (optional)	
Country	US
Validity Period	365 Days

Figure 12-34 Create a self-signed certificate

The obtained self-signed certificate is shown in Figure 12-35.



Key information for [bidard]

bidard

Key Size: 1024
Certificate Property:
Version X509 V3
Serial Number: 3F:32:A5:7E
Issued to:
bidard
itso
ibm
US
Issued by:
bidard
itso
ibm
US
Validity: Valid from August 6, 2003 to August 6, 2004
Fingerprint (MD5 Digest):
20:8D:B7:03:59:AE:AA:91:9C:9F:06:2D:DA:6C:DC:7D
Signature Algorithm: 1.2.840.113549.1.1.4(1.2.840.113549.1.1.4)

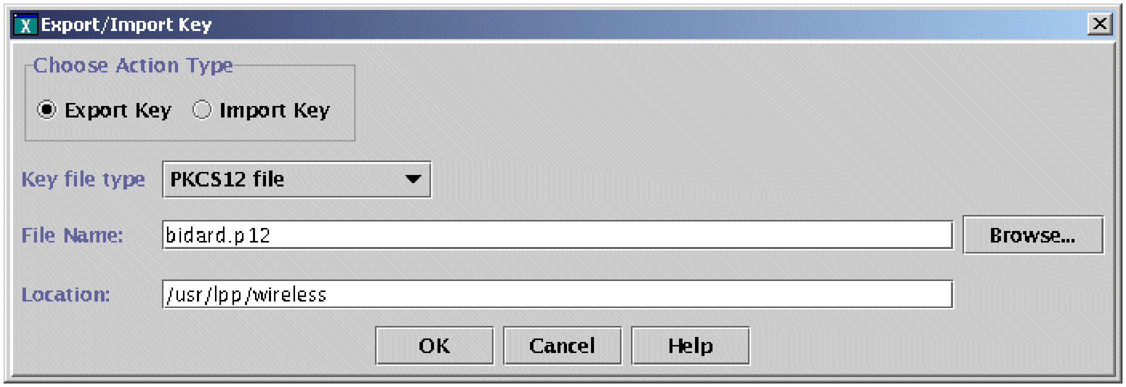
Error occured while getting NLS message for GUI_BUTTON_LABEL_VIEW_DETAILS

□ Set the certificate as the default

OK

Figure 12-35 Self-signed certificate view

The self-signed certificate subject name is cn=bidard,ou=itso,o=ibm,c=us and the key can be exported (see Figure 12-36) to be downloaded to the client.



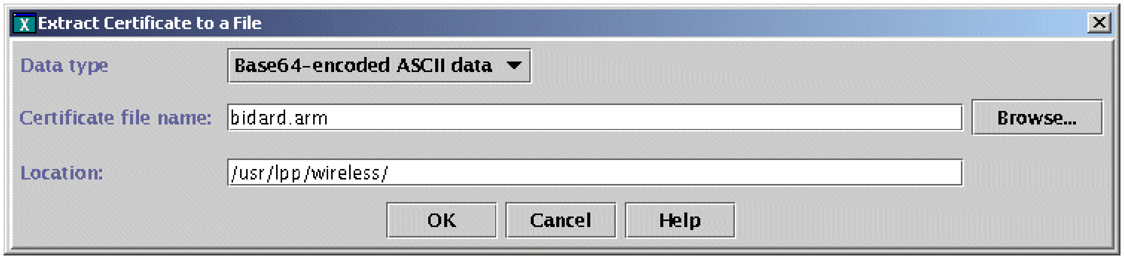
Export/Import Key

Choose Action Type
● Export Key ○ Import Key

Key file type: PKCS12 file
File Name: bidard.p12
Location: /usr/lpp/wireless

Figure 12-36 Export key

The self-signed certificate can also be extracted as shown in Figure 12-37.



Extract Certificate to a File

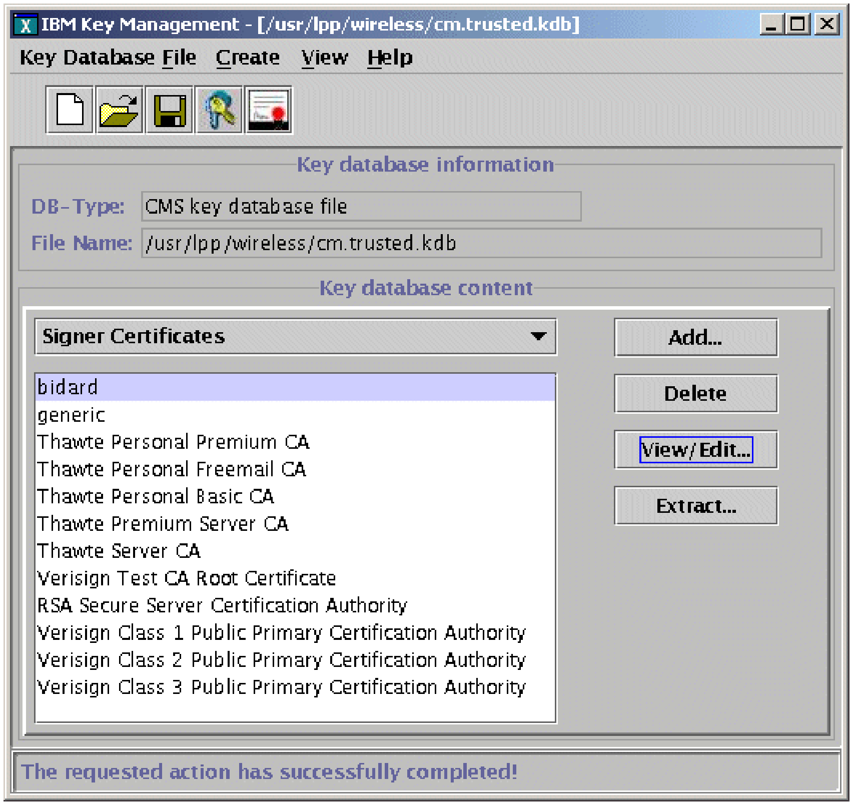
Data type: Base64-encoded ASCII data

Certificate file name: bidard.arm

Location: /usr/lpp/wireless/

Figure 12-37 Extract certificate

The certificate can be added to the signer view as shown in Figure 12-38.



IBM Key Management - [/usr/lpp/wireless/cm.trusted.kdb]

Key database information

DB-Type: CMS key database file

File Name: /usr/lpp/wireless/cm.trusted.kdb

Key database content

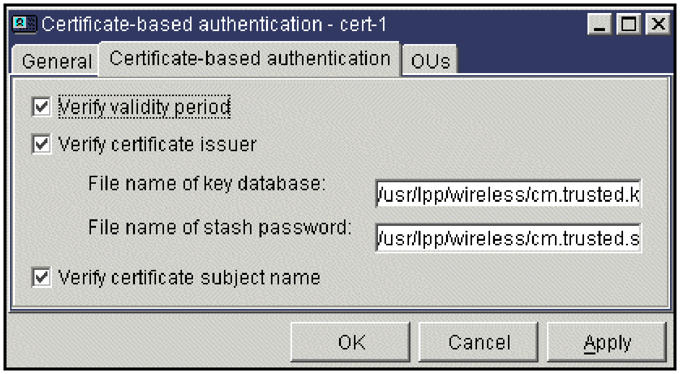
Signer Certificates

bidard
generic
Thawte Personal Premium CA
Thawte Personal Freemail CA
Thawte Personal Basic CA
Thawte Premium Server CA
Thawte Server CA
Verisign Test CA Root Certificate
RSA Secure Server Certification Authority
Verisign Class 1 Public Primary Certification Authority
Verisign Class 2 Public Primary Certification Authority
Verisign Class 3 Public Primary Certification Authority

The requested action has successfully completed!

Figure 12-38 ikeyman signer certificate view

You create a certificate based authentication profile.


Certificate-based authentication - cert-1
General
Certificate-based authentication
OUs
OK
Cancel
Apply

☑ Verify validity period

☑ Verify certificate issuer

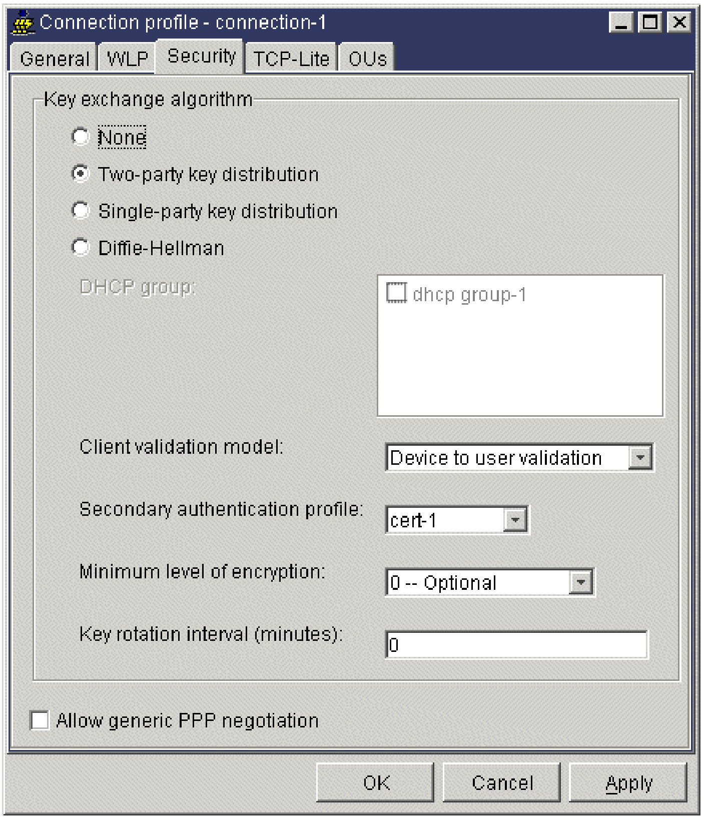
File name of key database: /usr/lpp/wireless/cm.trusted.k

File name of stash password: /usr/lpp/wireless/cm.trusted.s

☑ Verify certificate subject name

Figure 12-39 Certificate based authentication profile

Select the authentication algorithm in the connection profile. See Figure 12-39.



Key exchange algorithm

○ None

● Two-party key distribution

○ Single-party key distribution

○ Diffie-Hellman

DHCP group: □ dhcp group-1

Client validation model: Device to user validation

Secondary authentication profile: cert-1

Minimum level of encryption: 0 -- Optional

Key rotation interval (minutes): 0

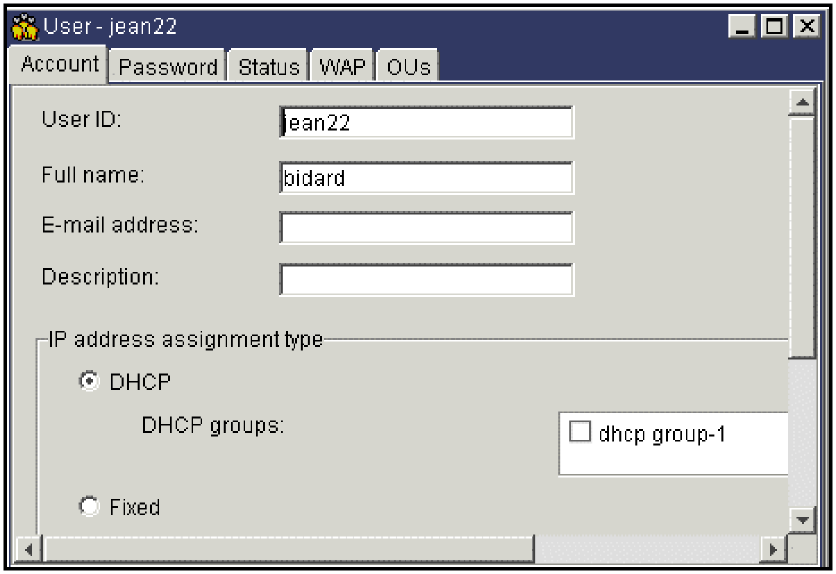
□ Allow generic PPP negotiation

Figure 12-40 Certificate based connection profile

To authenticate the user, the subject name should match the fully qualified distinguished name. It may also match just a subset of it:

- Its DN should contain the country (c=...) parameter.
- The LDAP entry should contain a matching Common Name (cn=...), User Id (UID=...) or SurName (sn=...) attribute.

Connection Manager server saves the user full name within the user LDAP entry CN and SN attributes.



User ID: jean22

Full name: bidard

E-mail address:

Description:

IP address assignment type

● DHCP

DHCP groups: □ dhcp group-1

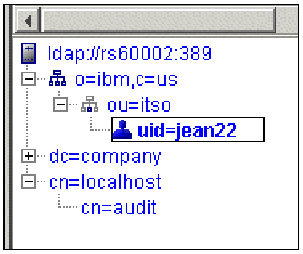
○ Fixed

Figure 12-41 User profile

The user LDAP server view shows the user DN as:

```
uid=jean22,ou=itso,o=ibm,c=us
```

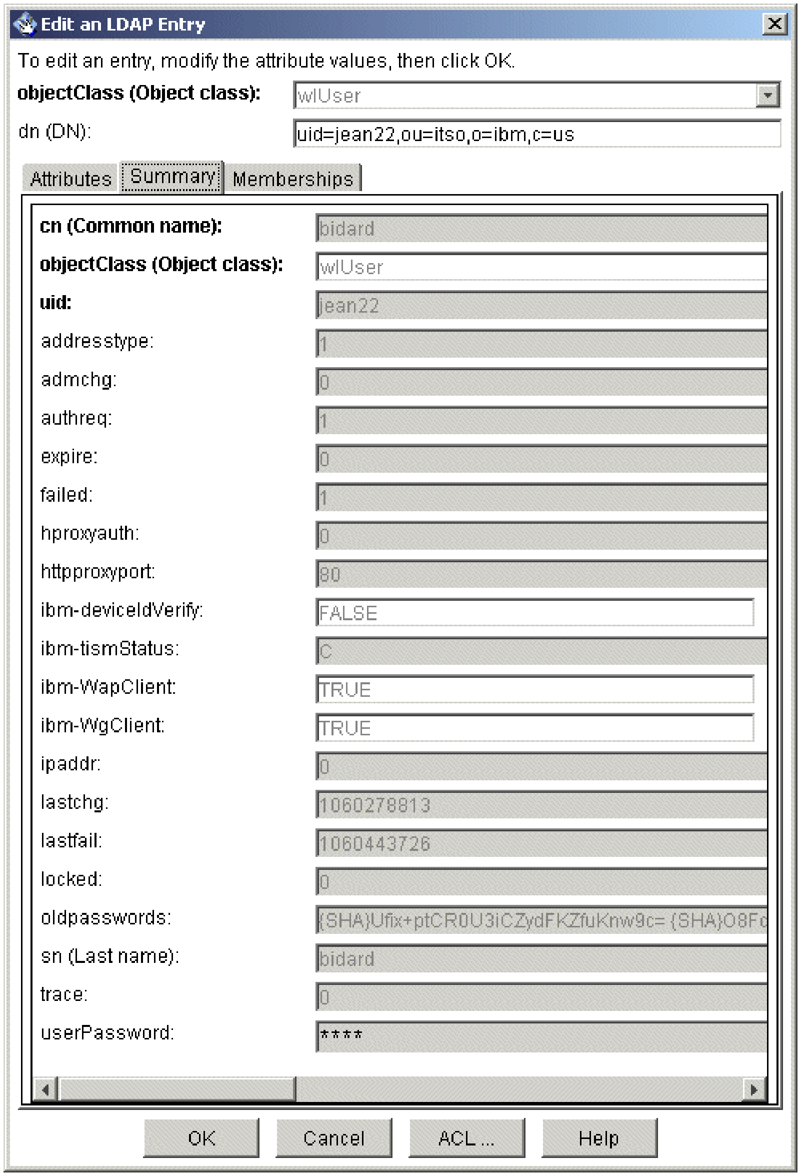
:



ldap://rs60002:389
o=ibm,c=us
ou=itso
uid=jean22
dc=company
cn=localhost
cn=audit

Figure 12-42 User LDAP view

This user LDAP entry has its CN and SN attributes equals to bidard.



Edit an LDAP Entry

To edit an entry, modify the attribute values, then click OK.

objectClass (Object class):	wlUser
dn (DN):	uid=jean22,ou=itso,o=ibm,c=us

cn (Common name):	bidard
objectClass (Object class):	wlUser
uid:	jean22
addresstype:	1
admchg:	0
authreq:	1
expire:	0
failed:	1
hproxyauth:	0
httpproxyport:	80
ibm-deviceIdVerify:	FALSE
ibm-tismStatus:	C
ibm-WapClient:	TRUE
ibm-WgClient:	TRUE
ipaddr:	0
lastchg:	1060278813
lastfail:	1060443726
locked:	0
oldpasswords:	{SHA}Ufix+ptCR0U3iCZydFKZfuKnw9c= {SHA}O8Fc
sn (Last name):	bidard
trace:	0
userPassword:	****

Figure 12-43 User LDAP entry

The connection setup challenges the Mobility Client to send a certificate as shown in Figure 12-44.

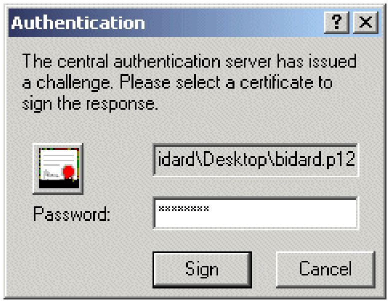
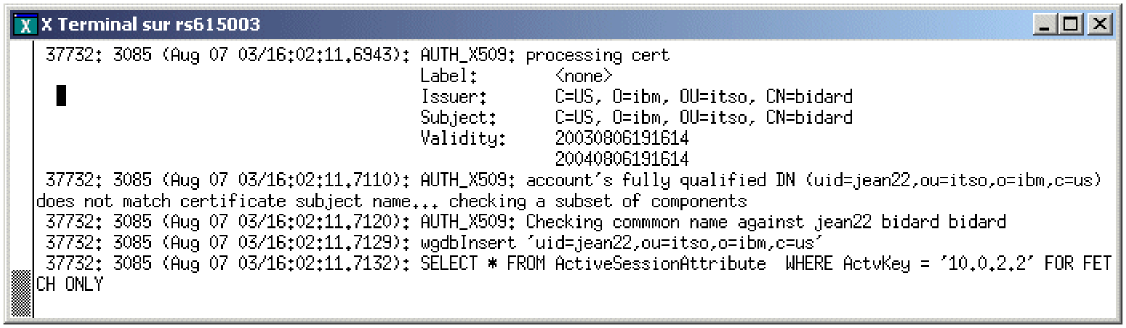

Authentication
?
The central authentication server has issued a challenge. Please select a certificate to sign the response.
idard\Desktop\bidard.p12
Password:
****
Sign
Cancel

Figure 12-44 Authentication

The Connection Manager log view shows the certificate subject name to the user LDAP entry matching as shown in Figure 12-45.



```
 37732: 3085 (Aug 07 03/16:02:11.6943): AUTH_X509: processing cert
                                        Label:        <none>
                                        Issuer:       C=US, O=ibm, OU=itso, CN=bidard
                                        Subject:      C=US, O=ibm, OU=itso, CN=bidard
                                        Validity:     20030806191614
                                                      20040806191614
 37732: 3085 (Aug 07 03/16:02:11.7110): AUTH_X509: account's fully qualified DN (uid=jean22,ou=itso,o=ibm,c=us)
does not match certificate subject name... checking a subset of components
 37732: 3085 (Aug 07 03/16:02:11.7120): AUTH_X509: Checking common name against jean22 bidard bidard
 37732: 3085 (Aug 07 03/16:02:11.7129): wgdbInsert 'uid=jean22,ou=itso,o=ibm,c=us'
 37732: 3085 (Aug 07 03/16:02:11.7132): SELECT * FROM ActiveSessionAttribute  WHERE ActvKey = '10.0.2.2' FOR FET
CH ONLY
```

Figure 12-45 wg.log certificate based authentication view

12.5 Wireless Transport Layer Security (WTLS) for WLP

Wireless Transport Layer Security (WTLS) for WLP has been implemented in Connection Manager as a specific Mobile Network Connection (MNC), and it can be used in place of the standard Wireless Layer Protocol (WLP) MNC for extra security.

Note: WebSphere Everyplace Connection Manager supports Wireless Transport Layer Security (WTLS) for WLP on Solaris and Linux platforms only. AIX support is not available.

Figure 12-46 illustrates the implementation of Wireless Transport Layer Security (WTLS) for WLP in Connection Manager.

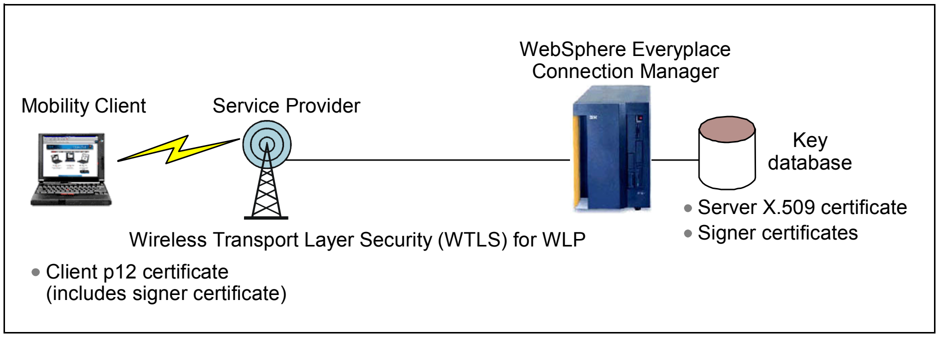

WebSphere Everyplace Connection Manager
Mobility Client
Service Provider
Key database
Server X.509 certificate
Signer certificates
Wireless Transport Layer Security (WTLS) for WLP
Client p12 certificate (includes signer certificate)

Figure 12-46 Wireless Transport Layer Security (WTLS) for WLP

WTLS for WLP was developed to enhance the security features already available in the Wireless Link Protocol (WLP) for Mobility Clients. It is an optional feature and if activated it will provide extra security support for the Mobility Client login and logoff procedures.

Note: The new WTLS for WLP is a different implementation of WTLS for WAP and WTLS for MQe.

In general, the WTLS for WLP implementation provides the following capabilities:

- Secure (encrypted) Mobility Client login and logoff procedures (connection setup)
- Triple DES (3DES) encryption is used.
- In addition to login procedure encryption, it can also be used for data encryption (3DES) when the proper configuration option is selected. If you want to encrypt data using this support, do not select the option **Only use WTLS for login or logoff sequences** as illustrated in Figure 12-48.

  When data encryption is active in WTLS, double encryption may occur. That is standard WLP encryption and in addition to WTLS encryption (3DES).

  Note: Standard WLP encryption supports DES, 3DES, and AES symmetric encryption algorithms while the new WTLS implementation supports 3DES only.
- Although WTLS does not provide user authentication, client, and server certificates, signers can optionally be verified (configuration option).
- As illustrated in Figure 12-46, server (WECM) requires a X.509 certificate and the Mobility Client requires a X.509 certificate in p12 (pfx) format.

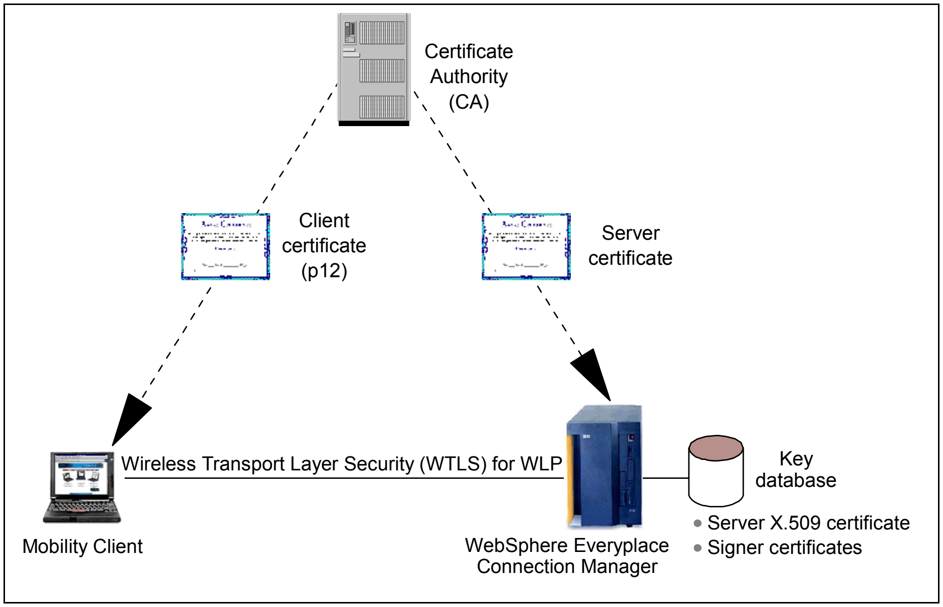

Certificate Authority (CA)
Client certificate (p12)
Server certificate
Wireless Transport Layer Security (WTLS) for WLP
Key database
Mobility Client
WebSphere Everyplace Connection Manager
• Server X.509 certificate
• Signer certificates

Figure 12-47 Mobility Client and Connection Manager certificates

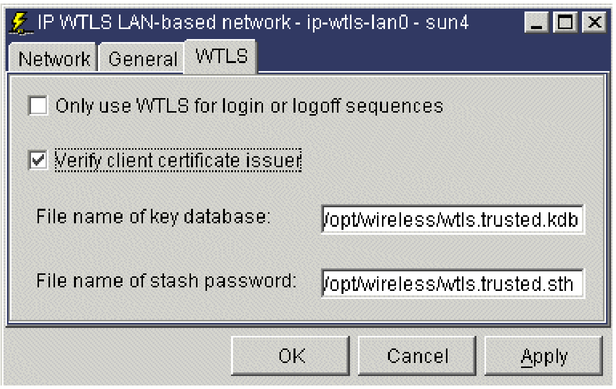
- After certificates have been sent and verified if required (configuration option), a random key is generated and exchanged for 3DES symmetric encryption of the login, logoff, and optional data (configuration option) procedures.

  Note: In this implementation of WTLS, there is no negotiation of the symmetric encryption algorithm or key length since 3DES is always used.

- The server (WECM) may check the Mobility Client certificate issuer with a root certificate saved in its key database (wtls.trusted.kdb by default).

  Note: You should also provide the file name of the key database and the file name of the stash password (see Figure 12-48).

- If you want Connection Manager to verify the Mobility Client certificate issuer (CA), select the box **Verify client certificate issuer** as shown in Figure 12-48.



□ Only use WTLS for login or logoff sequences

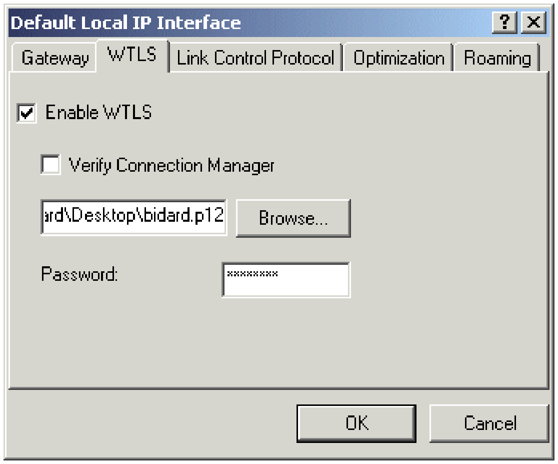
☑ Verify client certificate issuer

File name of key database: /opt/wireless/wtls.trusted.kdb

File name of stash password: /opt/wireless/wtls.trusted.sth

Figure 12-48 Server WTLS MNC configuration

- As an option, the Mobility Client may also check the server certificate against its p12 signer certificate. For this reason, when this option is used, both the Mobility Client p12 certificate (includes its signer certificate) and the server certificate must be issued by the same signer root or CA.



☑ Enable WTLS

□ Verify Connection Manager

ard\Desktop\bidard.p12

Password: ********

Figure 12-49 Client WTLS network property

13

MQe application traffic optimization

The objective of this chapter is provide information on how to make full use of the optimization features of MQe and Connection Manager in order to reduce the number of bytes and packets transmitted over a network.

13.1 Overview

The advent of GPRS and the popularity of packet cellular networks has led to the introduction of different pricing models for network usage such as charging per volume of data transmitted. This has resulted in an increased interest from users of network services in the reduction of the cost of their network charges by reducing the overheads required to send data over a network. MQe and Connection Manager are designed with this type of requirement in mind, and enable considerable optimizations to be performed on data flow over a network.

Figure 13-1 illustrates the environment used to perform the tasks used to quantify the optimizations described in this chapter.

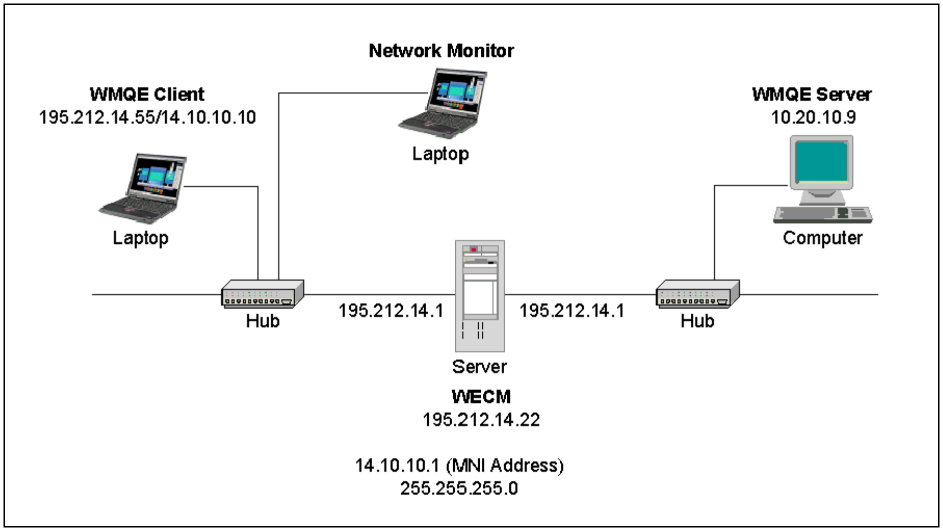


Figure 13-1 Sample scenario

A private Ethernet network was set up with Connection Manager configured as a gateway between a MQe client and MQe server. Two Ethernet hubs were installed, one on each side of the Connection Manager gateway in order to enable network monitoring.

Connection Manager version 5.0.0.1 was installed on AIX version 5.2. The Connection Manager Mobility Client, version 5.0.0.1 was installed on the laptop. The Connection Manager Mobility Client enables a VPN to be established between the client and the Connection Manager gateway, and is required in order to take full advantage of the optimization features of Connection Manager. This is a somewhat un-natural configuration of having a VPN through a one hop network, but is a sufficient model to evaluate the optimizations of concern.

MQe version 2.0.0.6 was installed on a Windows 2000 PC and a Windows 2000 laptop. MQe client and server applications were developed to send and receive MQe messages over Connection Manager. The server was installed on the PC, the client on the laptop.

13.1.1 IP configuration

Two Ethernet adapters were configured in the Connection Manager AIX server, one for the client side and one for the server side.

en0:

- IP address:10.20.10.1
- Subnet mask:255.255.255.0
- Default gateway:195.212.14.22

en1:

- IP address:195.212.14.19
- Subnet mask:255.255.255.0
- Default gateway:195.212.14.22

The laptop Ethernet adapter was configured for the client side of the network:

- 195.212.14.55
- 255.255.255.0
- 195.212.14.1

The PC Ethernet adapter was configured for the server side of the network:

- 10.20.10.9
- 255.255.255.0
- 10.20.10.1

13.1.2 Connection Manager gateway initial configuration

After installing Connection Manager, the Connection Manager Gatekeeper Administration tool was used to configure the resources required for the VPN. This included the following components:

- **Connection Manager**: This is the fundamental resource that defines the gateway for which all other resources are defined.
- **Mobile Network Connection**: This is the component of Connection Manager that accesses the bearer network, in this case the Ethernet network.
- **Mobile Network Interface**: This resource defines a range of IP addresses used to form the virtual network.

  - **Connection Profile Resource**: This is used to set the control parameters of the chosen optimizations
  - **TCP-Lite Service**: A new Connection Manager feature, which maps TCP transport protocol into UDP transport while preserving the end-to-end client/server TCP connection and delivery acknowledgment and retransmission guarantee.
  - **Connection Manager Defined Users**: This is used to connect to the Connection Manager gateway from a Connection Manager Mobility Client.

The MNI was configured to define an IP subnetwork:

- IP address: 14.10.10.1
- Subnet mask: 255.255.255.0

13.1.3 Connection Manager Mobility Client configuration

The Connection Manager Mobility Client was configured on the laptop by creating a basic IP bearer *connection* definition to the Connection Manager gateway. When creating the connection definition it is necessary to define:

- The name of the connection
- Whether to use a backup connection (select **No**)
- The type of network over which the connection is made, in this case IP, WiFi, GPRS, 1xRTT, Broadband
- The IP address of the Connection Manager gateway: 195.212.14.19
- The specific network interface over which the connection will be made; a number of network interfaces may be specified in order to enable roaming. For this exercise a single interface was selected: **Default Local IP Interface**. In a non laboratory environment we suggest the use of a specific IP interface.

13.1.4 Routing tables

Before the VPN connection was established between the Connection Manager client on the laptop and the Connection Manager gateway, the routing tables were as follows:

Connection Manager gateway

```
Routing tables

Destination groups gateway     Flags   Refs    Use  If    PMTU  Exp

Route Tree for Protocol Family 2 (Internet):
default         195.212.14.22     UGc       0        0  en1     -   -
```

```
10.20.10/24      10.20.10.1        U          0         0  en0      -   -
10.20.10.1       127.0.0.1         UGHS       0         0  lo0      -   -
14.10.10/24      14.10.10.1        U          0         0  mn0      -   -
14.10.10.1       127.0.0.1         UGHS       0        32  lo0      -   -
127/8            127.0.0.1         U         46      1911  lo0      -   -
195.212.14/24    195.212.14.19     U          0         0  en1      -   -
195.212.14.19    127.0.0.1         UGHS       0         0  lo0      -   -
```

Laptop

Active routes:

```
Network destination      Netmask          Gateway        Interface    Metric
          0.0.0.0          0.0.0.0     195.212.14.1   195.212.14.55       1
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1       1
     195.212.14.0    255.255.255.0    195.212.14.55   195.212.14.55       1
    195.212.14.55  255.255.255.255        127.0.0.1       127.0.0.1       1
   195.212.14.255  255.255.255.255    195.212.14.55   195.212.14.55       1
        224.0.0.0        224.0.0.0    195.212.14.55   195.212.14.55       1
  255.255.255.255  255.255.255.255    195.212.14.55               2       1
```

Default gateway: 195.212.14.1

MQe server

Active routes:

```
Network destination      Netmask          Gateway        Interface    Metric
          0.0.0.0          0.0.0.0       10.20.10.1      10.20.10.9       1
       10.20.10.0    255.255.255.0       10.20.10.9      10.20.10.9       1
       10.20.10.9  255.255.255.255        127.0.0.1       127.0.0.1       1
   10.255.255.255  255.255.255.255       10.20.10.9      10.20.10.9       1
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1       1
        224.0.0.0        224.0.0.0       10.20.10.9      10.20.10.9       1
  255.255.255.255  255.255.255.255       10.20.10.9      10.20.10.9       1
```

Default gateway: 10.20.10.1

After the Connection Manager Client on the laptop had established a connection with the Connection Manager gateway, the laptop had additional routes pushed down to it. The reason for this is nature of our IP address choices. It also reflects the fact that the Connection Manager virtual network is not the default route on the client. In order to force the MQe traffic through the VPN, the 10.20.0.0 network route is added to the MNI as a *negotiated route*. In general, a return network route is needed to be set on the MQe server to get back into the virtual network. In our case we relied on the default route on the MQe server being the Connection Manager machine's physical address.

Active routes:

Network destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	195.212.14.1	195.212.14.55	1
10.20.10.0	255.255.255.0	14.10.10.1	14.10.10.10	1
14.10.10.0	255.255.255.0	14.10.10.10	14.10.10.10	1
14.10.10.10	255.255.255.255	127.0.0.1	127.0.0.1	1
14.255.255.255	255.255.255.255	14.10.10.10	14.10.10.10	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
195.212.14.0	255.255.255.0	195.212.14.55	195.212.14.55	1
195.212.14.55	255.255.255.255	127.0.0.1	127.0.0.1	1
195.212.14.255	255.255.255.255	195.212.14.55	195.212.14.55	1
224.0.0.0	224.0.0.0	14.10.10.10	14.10.10.10	1
224.0.0.0	224.0.0.0	195.212.14.55	195.212.14.55	1
255.255.255.255	255.255.255.255	14.10.10.10	14.10.10.10	1

Default gateway: 195.212.14.1

PING tests

When not connected to Connection Manager, the laptop can successfully ping the following IP address:

```
    195.212.14.19
```

The PC can ping these IP addresses:

```
    10.20.10.1
    195.212.14.19
```

Once the Connection Manager client on the laptop had connected to Connection Manager and a VPN was created; the laptop was assigned the following additional IP parameter:

```
address - 14.10.10.10
network route – 10.20.10.0
```

The laptop can then additionally ping the following IP addresses:

```
10.20.10.1
all IP addresses in the server network including 10.20.10.9
```

The PC could additionally ping:

```
195.212.14.55
14.10.10.10
```

13.1.5 Connection Manager settings

The following diagram shows the Connection Manager resources used to configure and optimize the VPN and the relationships between them. It shows the main resources and parameters that were configured to optimize network traffic.

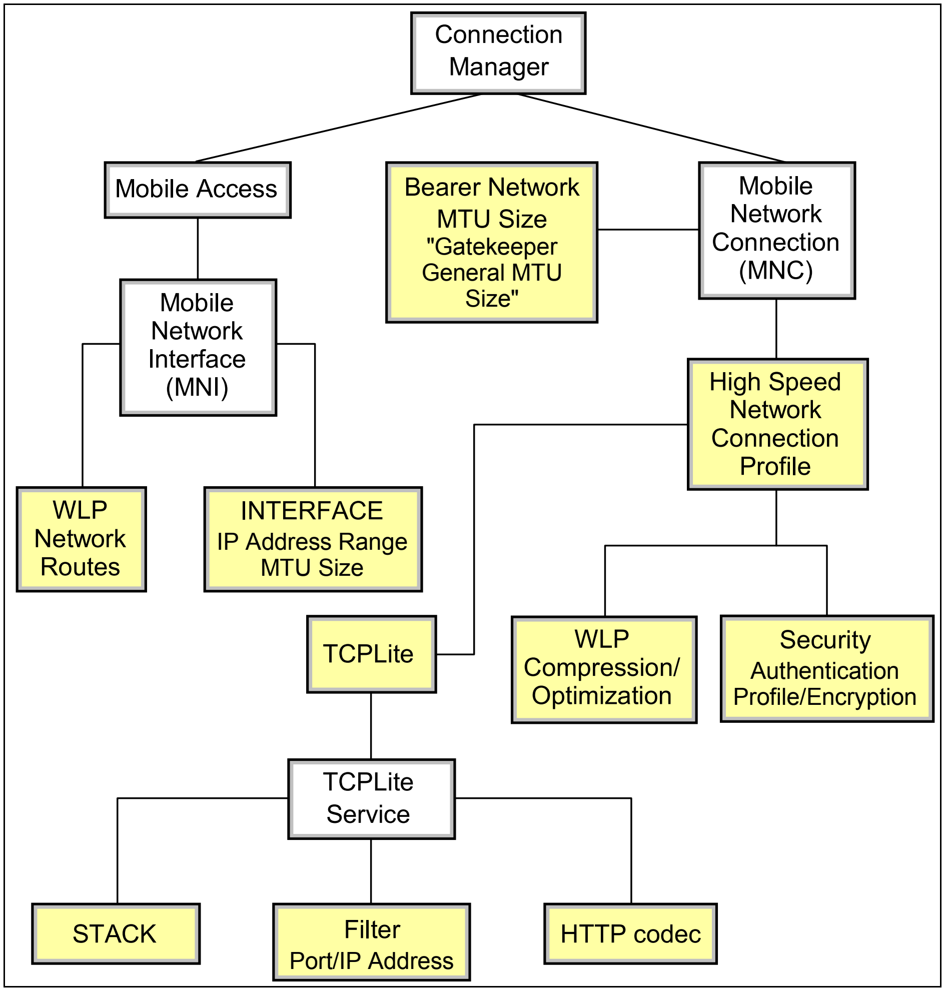

Connection Manager
Mobile Access
Bearer Network MTU Size "Gatekeeper General MTU Size"
Mobile Network Connection (MNC)
Mobile Network Interface (MNI)
High Speed Network Connection Profile
WLP Network Routes
INTERFACE IP Address Range MTU Size
TCPLite
WLP Compression/ Optimization
Security Authentication Profile/Encryption
TCPLite Service
STACK
Filter Port/IP Address
HTTP codec

Figure 13-2 Connection Manager resources and options

The white boxes are the main Connection Manager resources configured for the VPN. The yellow boxes show the properties of those resources with the uppercase italicized items showing the name of the tabs of the properties pages.

Figure 13-3 shows these resources displayed in the Connection Manager Gatekeeper Administration interface.

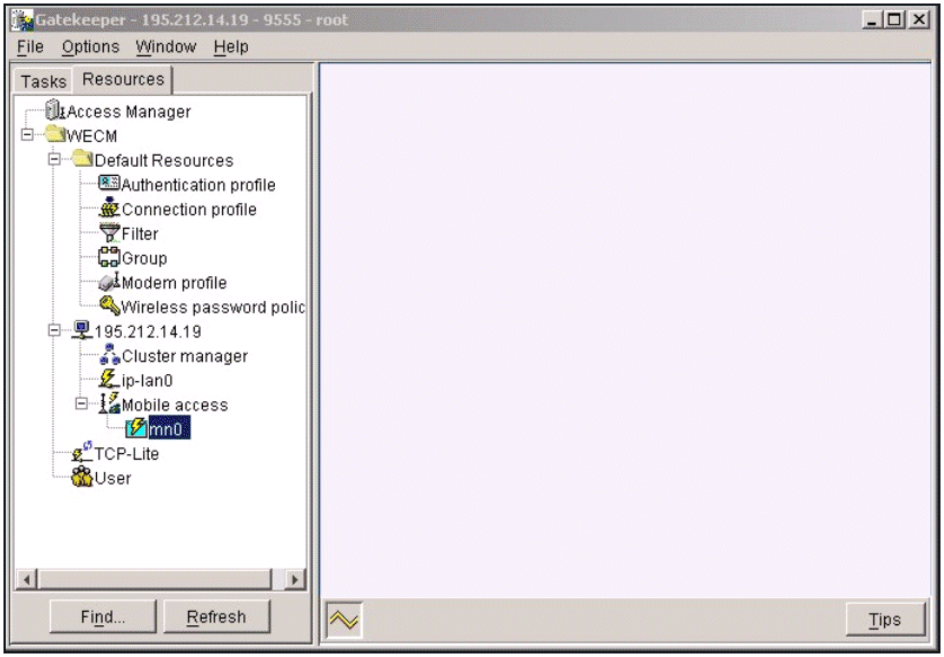
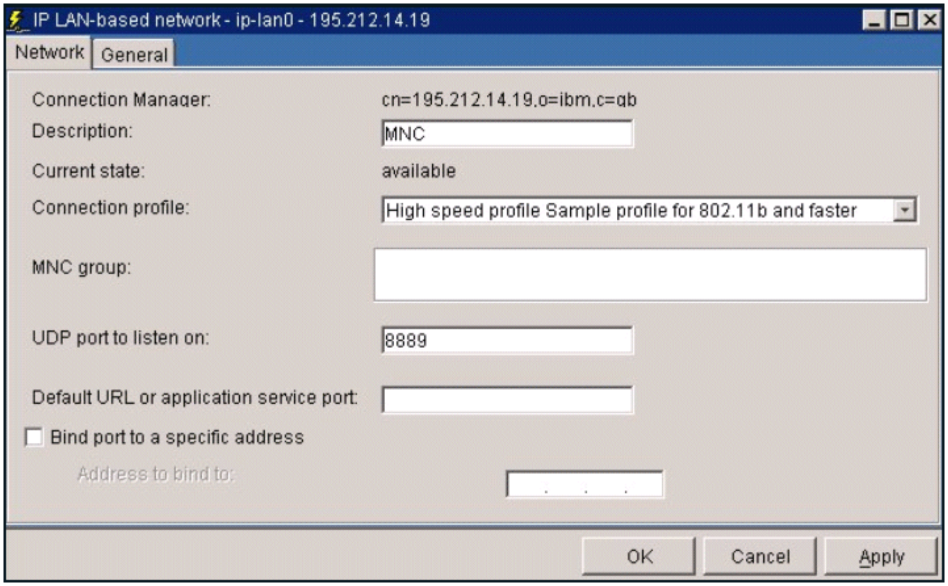


Figure 13-3 Connection Manager resources

Mobile Network Connection (MNC)

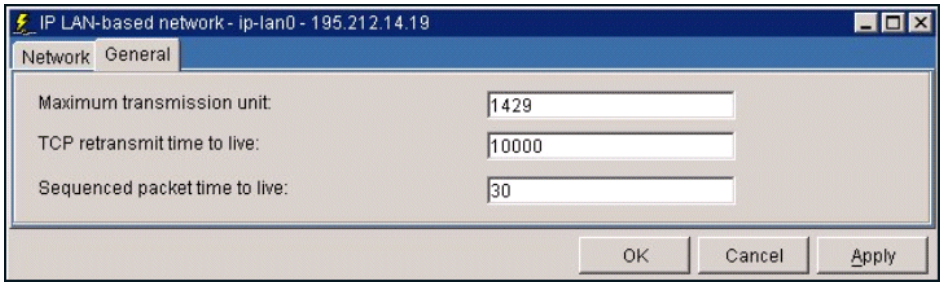
On the Network tab of the MNC properties, a connection profile must be selected for the MNC from the list of defined connection profiles. A number of connection profiles are predefined during the installation of Connection Manager including the high speed profile. This profile is appropriate for Ethernet and other high bandwidth networks.



Connection Manager: cn=195.212.14.19,o=ibm,c=gb
Description: MNC
Current state: available
Connection profile: High speed profile Sample profile for 802.11b and faster
MNC group:
UDP port to listen on: 8889
Default URL or application service port:
□ Bind port to a specific address
Address to bind to: . . .

Figure 13-4 Mobile Network Connection (MNC)

The Maximum Transmission Size (MTU) value for the MNC can be modified on the General tab of the MNC properties. Since we are using TCP-Lite the TCP retransmit time to live will be superseded by TCP-Lite control parameters. The MTU of 1429 was discovered empirically based upon IP fragmentation in the physical network. A network sniffer tool was used to observe IP fragmentation behavior. As we are using the TCP-Lite transport, the TCP retransmit time to live is not invoked, similarly with the sequenced packet time to live.

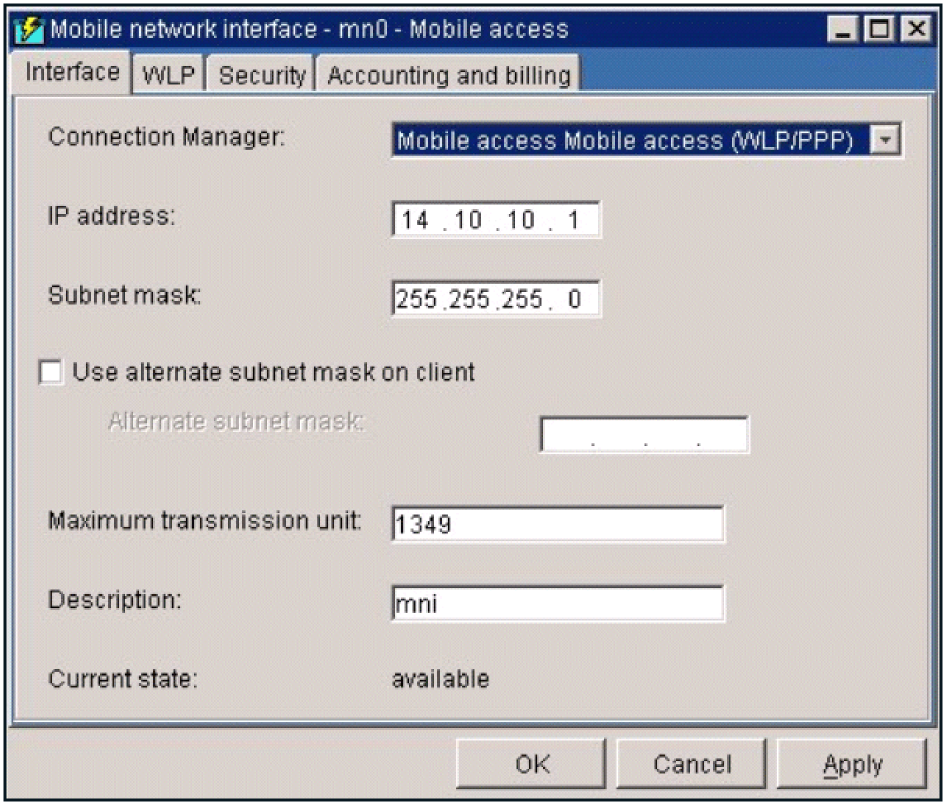


Maximum transmission unit: 1429
TCP retransmit time to live: 10000
Sequenced packet time to live: 30

Figure 13-5 Maximum transmission unit

Mobile Network Interface (MNI)

The Interface tab of the MNI enables the IP address range to be defined, as well as the MTU value. The MTU on this resource is the MTU that the gateway's virtual interface presents to the host IP stack. Since the gateway MNI is not a participant in maximum segment size determination, the MNI MTU is somewhat irrelevant. Empirical study of MNI MTU bears this out.



Connection Manager: Mobile access Mobile access (WLP/PPP)

IP address: 14 . 10 . 10 . 1

Subnet mask: 255 .255 .255 . 0

☐ Use alternate subnet mask on client

Alternate subnet mask: . . .

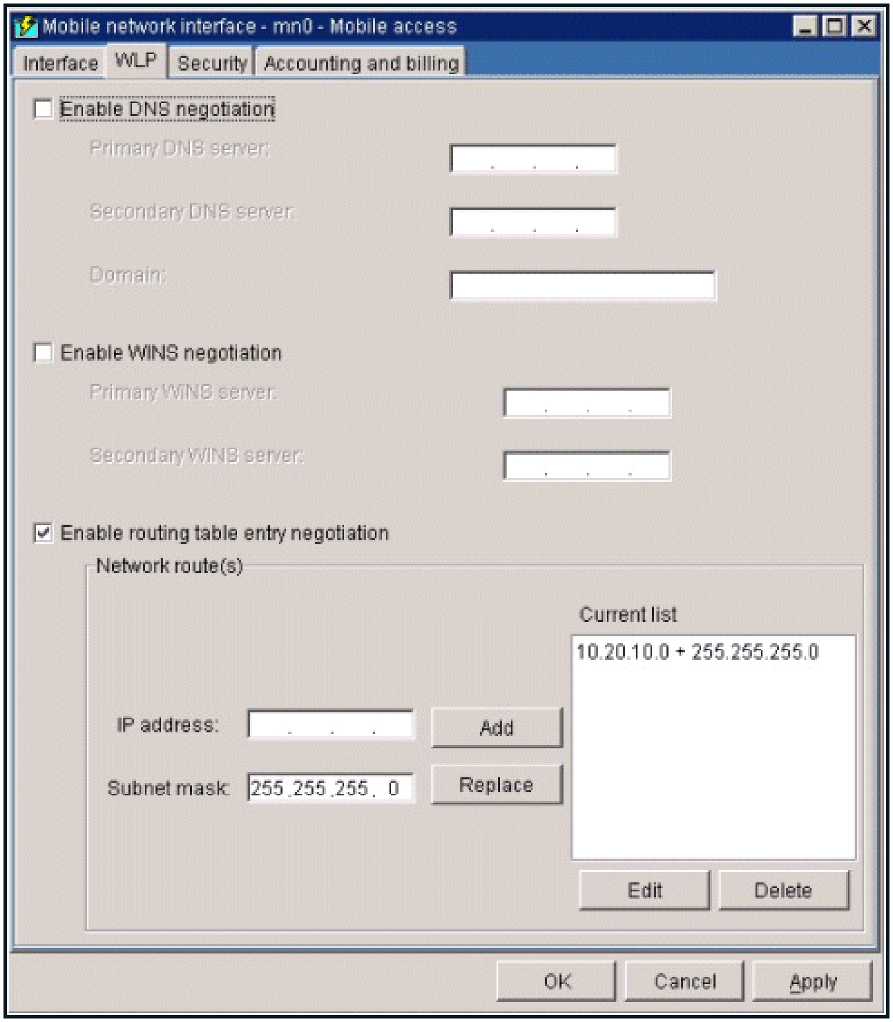
Maximum transmission unit: 1349

Description: mni

Current state: available

Figure 13-6 Mobile Network Interface (MNI)

On the Wireless Link Protocol (WLP) page, additional routes may be defined which will be pushed down to Mobility Clients when they connect. See Figure 13-7; visible here is the network route to the MQe server.



□ Enable DNS negotiation

Primary DNS server:

Secondary DNS server:

Domain:

□ Enable WINS negotiation

Primary WINS server:

Secondary WINS server:

☑ Enable routing table entry negotiation

IP address:

Subnet mask: 255.255.255. 0

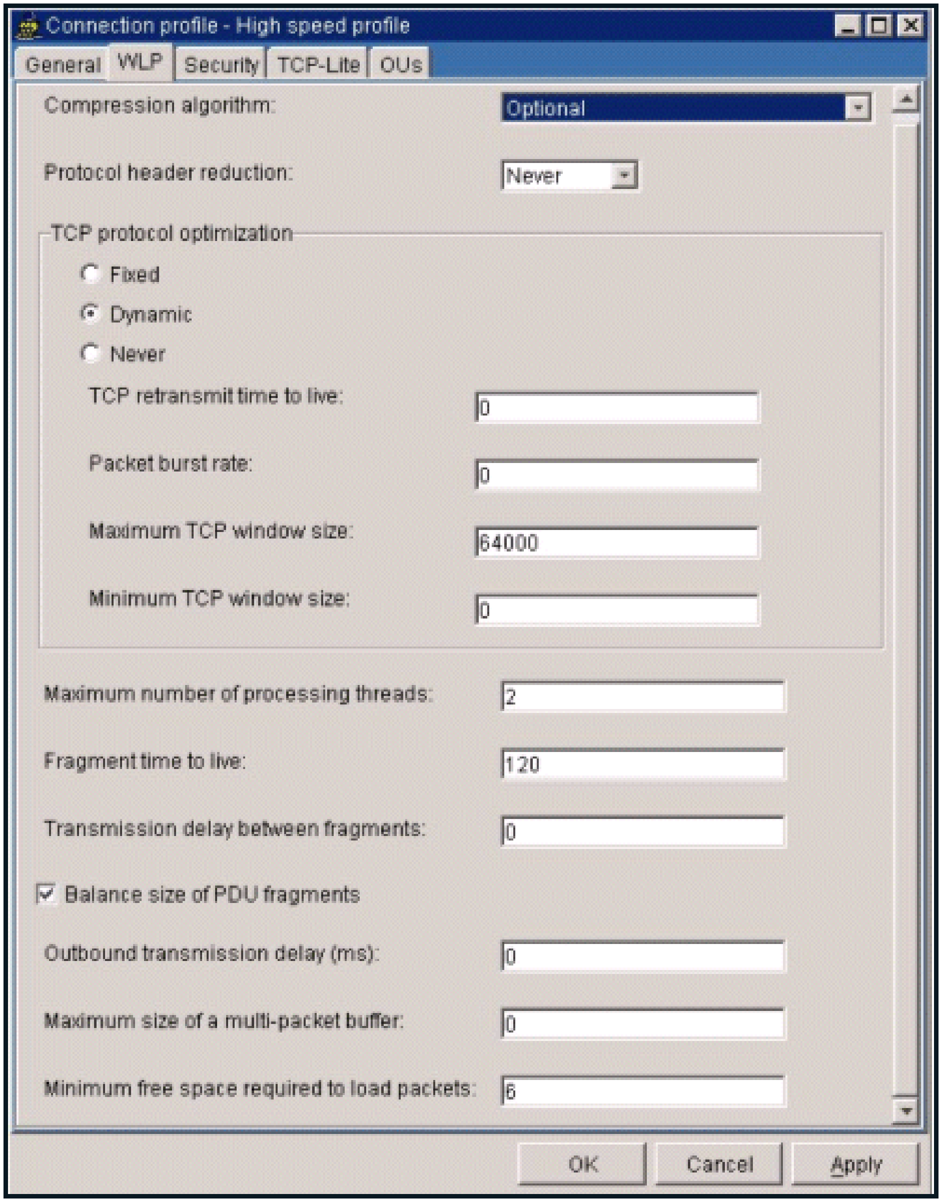
Current list

10.20.10.0 + 255.255.255.0

Figure 13-7 Network routes

13.1.6 Connection profile

The Wireless Link Protocol (WLP) page enables compression and optimization settings to be modified. Here too since we are using TCP-Lite the TCP protocol optimization will be superseded, as will some of the remaining WLP parameters shown on this screen shot. See the MQe5002 stack screen shot for the parameters that are superseded.



Connection profile - High speed profile

Compression algorithm: Optional

Protocol header reduction: Never

TCP protocol optimization

○ Fixed

● Dynamic

○ Never

TCP retransmit time to live: 0

Packet burst rate: 0

Maximum TCP window size: 64000

Minimum TCP window size: 0

Maximum number of processing threads: 2

Fragment time to live: 120

Transmission delay between fragments: 0

☑ Balance size of PDU fragments

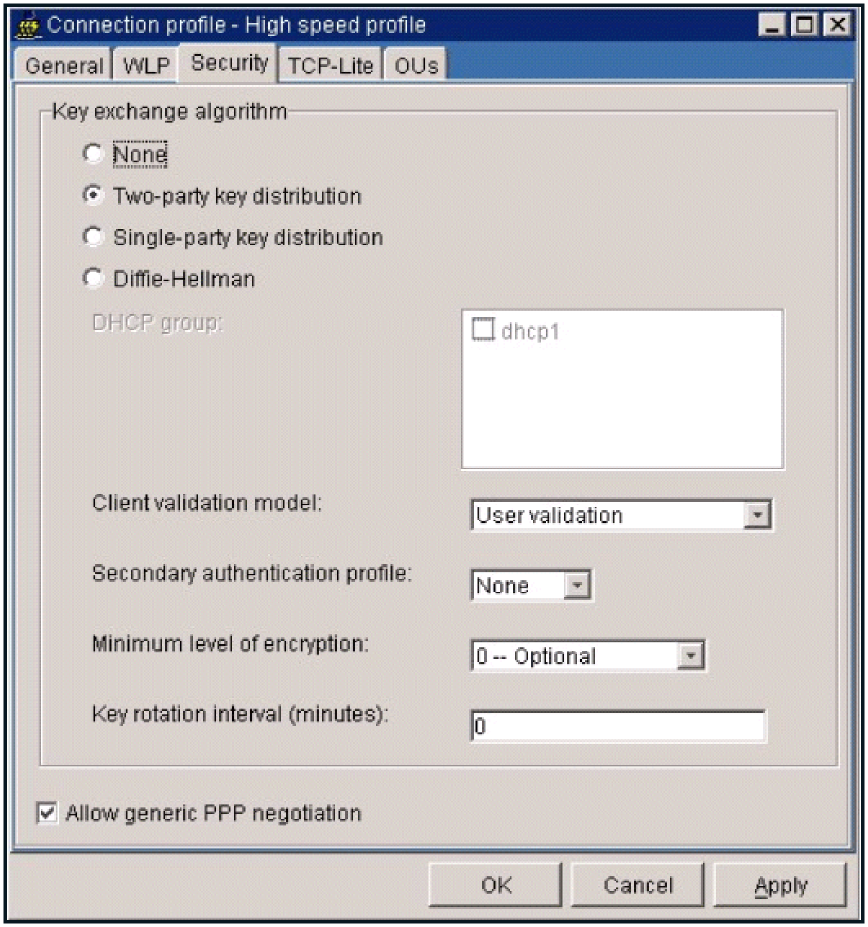
Outbound transmission delay (ms): 0

Maximum size of a multi-packet buffer: 0

Minimum free space required to load packets: 6

Figure 13-8 Wireless Link Protocol (WLP) configuration

The security page enables authentication and encryption settings to be modified; we chose to accept the defaults.



Key exchange algorithm

○ None
● Two-party key distribution
○ Single-party key distribution
○ Diffie-Hellman

DHCP group: □ dhcp1

Client validation model: User validation

Secondary authentication profile: None

Minimum level of encryption: 0 -- Optional

Key rotation interval (minutes): 0

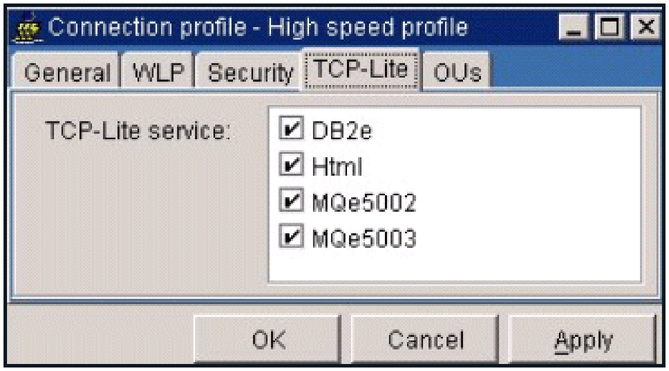
☑ Allow generic PPP negotiation

Figure 13-9 Security configuration

The TCP-Lite page lists any TCP-Lite services that have been defined. A TCP-Lite service is enabled by first creating a TCP-Lite object by adding a TCP-Lite resource, and then selecting it in a connection profile that is assigned to your MNC.

The following picture shows the TCP-Lite services built for this exercise. The following screen shots for TCP-Lite are taken from the MQe5002 TCP-Lite object. The services are differentiated by destination TCP port. TCP-Lite is symmetric, and cares not whether the TCP client sits on the Connection Manager client or up in the enterprise LAN.

It is worth noting the HTML object shown in Figure 13-10 includes an HTTP codec. TCP-Lite objects containing HTTP codecs require the transactions to be client initiated.



TCP-Lite service:
☑ DB2e
☑ Html
☑ MQe5002
☑ MQe5003

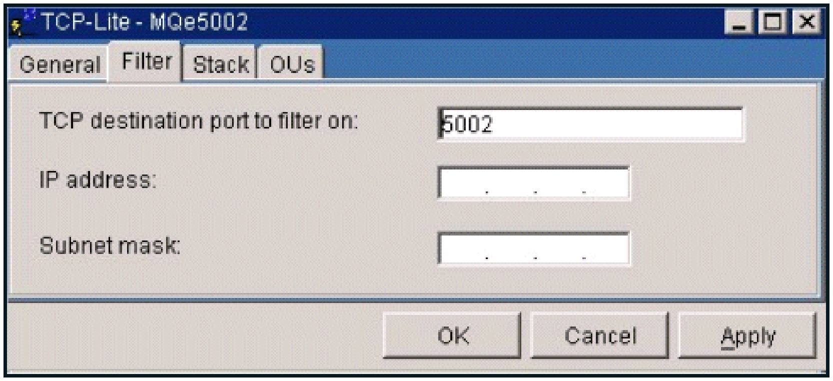
Figure 13-10 TCP-Lite service

13.1.7 TCP-Lite service

The TCP-Lite service can be applied to the following:

- A port number and/or,
- An IP address or,
- A range of IP addresses by specifying a subnet mask.

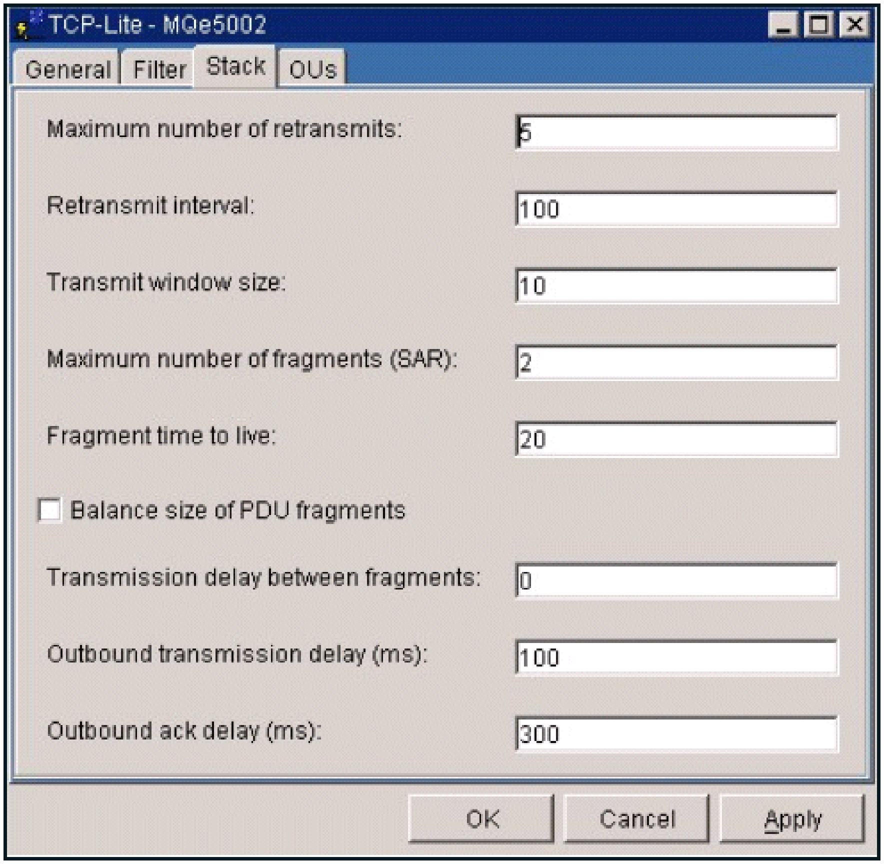
A TCP-Lite service has been defined for ports 5002 and 5003 as these are the ports used by the MQe application. The other two objects were used for work outside the scope of this scenario, and were configured to use ports other than 5002 and 5003.



TCP destination port to filter on: 5002
IP address: . . .
Subnet mask: . . .

Figure 13-11 TCP-Lite filter

TCP-Lite optimization settings are defined on the *Stack* page as illustrated in Figure 13-12.


TCP-Lite - MQe5002

Maximum number of retransmits: 5

Retransmit interval: 100

Transmit window size: 10

Maximum number of fragments (SAR): 2

Fragment time to live: 20

□ Balance size of PDU fragments

Transmission delay between fragments: 0

Outbound transmission delay (ms): 100

Outbound ack delay (ms): 300

Figure 13-12 TCP-Lite stack settings

13.1.8 Optimization techniques

Connection Manager utilizes a variety of optimization techniques, which may be tuned for a specific solution according to the requirements:

- Compression of the data payload
- TCP retransmission suppression and dynamic window size modification based on bandwidth product delay estimate. This is superseded by TCP-Lite.
- Packet joining: If the packets are small, they can be joined and sent as a single packet up to the value of the maximum transmission unit (MTU) for the network.

- IP header reduction: Compression of the TCP/IP headers. This is superseded by TCP-Lite.
- TCP-Lite encodes TCP headers and joins pure TCP control packets.
- HTTP codec encodes and HTTP header elements.

Optimization is performed at the network, transport, and application layers. Both the Connection Manager client and server have an implementation of a virtual device driver.

When running MQe over Connection Manager, optimal compression is achieved by using a compressor within MQe. This allows the entire message to be compressed in one stage. Connection Manager compression is not turned off and will only take place if further compression may be achieved at a packet level.

13.1.9 MTU sizes

An Maximum Transmission Unit (MTU) is the largest possible unit of data (in bytes) that can be sent in a single frame. MTUs are defined for MNCs and MNIs. The MNI is the virtual adapter, and the MNC is the interface between Connection Manager and the physical network.

To calculate the value of each MTU start by empirically learning the MTU of the bearer network. Then set the MNC MTU to bearer network MTU – 28. These 28 bytes are used by the IP stack when it performs UDP encapsulation of the packet after Connection Manager is done manipulating it, and hands it back to the stack. MNI MTU is presented to the IP stack as the virtual adapter’s MTU.

Once MNC MTU is known, we subtract different amounts depending upon the nature of the Connection Manager configuration and the application’s data. In the case where Connection Manager is configured with TCP-Lite and HTTP codec, the WLP header can be as much as 60 bytes. In the case where Connection Manager encrypts non-compressible data, that data can grow; the 60 bytes allocated for WLP header allows for this. This puts the MNI MTU at bearer network MTU - 28 – 60

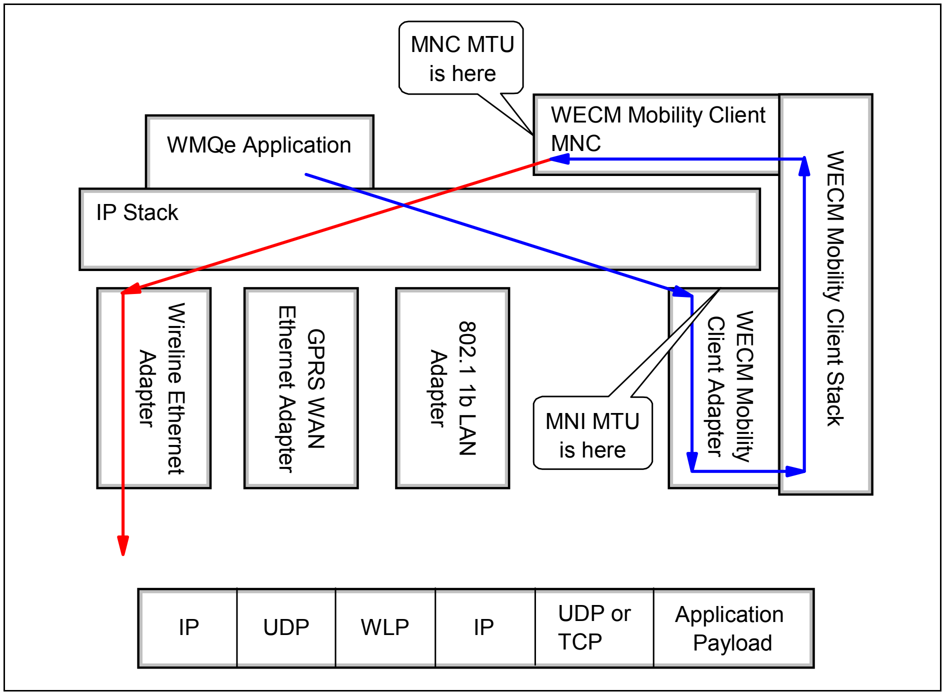
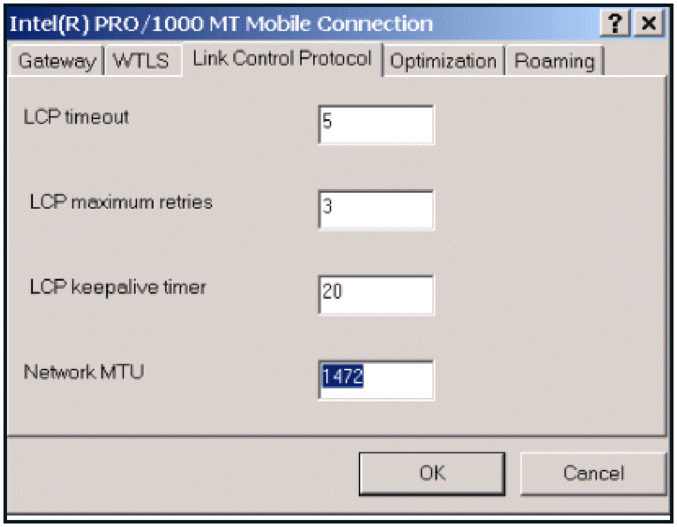


Figure 13-13 Mobility Client message flow

To set the Connection Manager gateway MNC MTU, from the Gatekeeper, right-click the **ip-lan0 resource** (the MNC) and select **Properties**. Select the **General** tab, update the MTU value and click **Ok**.

To set the Connection Manager gateway MNI MTU, open the **Properties** of the MNI (under the Mobile Access) resource. The MTU valued can be entered from the Interface tab.

The Connection Manager Client MTUs must be changed to the same values as the gateway. The MNC MTU is defined in the GUI from the connection definition's interface properties. Select the interface, the default IP interface in our case, then click the **Link Control Protocol** tab.



LCP timeout: 5

LCP maximum retries: 3

LCP keepalive timer: 20

Network MTU: 1472

Figure 13-14 Link Control Protocol - Network MTU

If you plan on using Connection Manager's roaming features you should consider defining each physical interface to be used. Since each interface is typically a different bearer, wireline LAN, WiFi, GPRS, each of these has its own control parameters, which will be reset at roam time.

To update the Connection Manager Client MNI MTU, the device driver must be updated:

1. Open the Windows Device Manager (in Windows 2000, this can be found in **Control Panel -> Administrative Tools -> Computer Management**).
2. Select **View -> Show Hidden Devices.**
3. Expand the **Network Adapters** item.
4. Find the IBM Mobility Interface for Windows, right-click it and select **Properties.**
5. Select the **Advanced** tab.
6. Select the MTU size in the Property list, and update the value as required.

MQe also enables the packet size to be specified for the communications adapter. This should be set to the same value as the Connection Manager MNI value, in our case 1349.

13.1.10 TCP-Lite settings

TCP-Lite is an optimization feature in Connection Manager, which provides a transport channel that intercepts TCP in order to reduce the overhead involved in session management. TCP-Lite encodes TCP headers, combines many encoded headers into single transmission, combines many payloads into single transmission, and generates retransmissions when required. The aggressive packet joining is accomplished through the introduction of configured artificial latency. The maintenance of the TCP end-to-end session for the application is accomplished by constructing a virtual version of the real endpoint peer in the application layer on the Connection Manager client and the Connection Manager gateway. The guaranteed delivery and sequencing of TCP is maintained while the retransmission interval and retransmission count are externalized as parameters of the TCP-Lite object.

A TCP-Lite service is created by right-clicking **Default Resources**, selecting **Add Resource**, then **TCP-Lite -> TCP-Lite**. A TCP-Lite service may be associated with a port number and IP address in order to tailor optimization to specific applications or servers. Once created, the TCP-Lite service is associated with an MNC by updating the TCP-Lite settings of the connection profile to use this service.

To display a list of the TCP-Lite services defined, double-click the **TCP-Lite resource**. To display the TCP-Lite properties for a specific service, right-click a service and select **Properties**. On the Stack tab of the properties window, a number of parameters may be modified for optimization.

There are two delays that can be set on TCP-Lite transmissions:

- Outbound transmission delay (milliseconds): This delay is applied to payload packets.
- Outbound ack delay (milliseconds): This delay is only applied to pure TCP control messages, for example, pure ACKs.

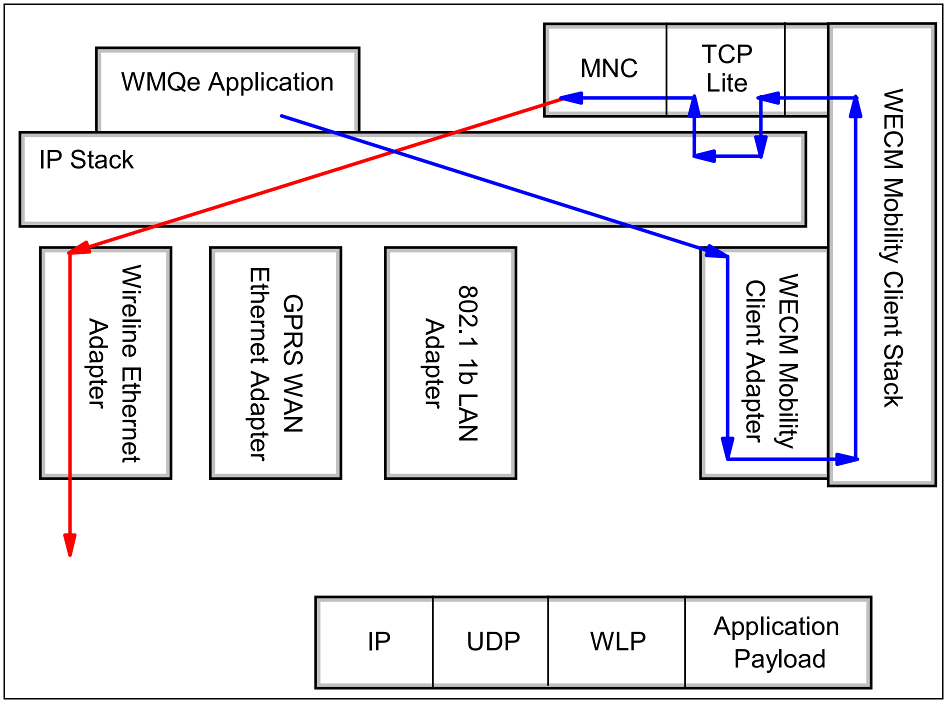
These delays enable Connection Manager to perform aggressive packet joining if possible to reduce the number of packets and bytes sent over the network.

Each delay must not be greater than the retransmit interval, which is defined in seconds. So, for a retransmit interval of 10, each delay must not be more than 10000, otherwise, this can cause collisions of the retransmission timer with the delay timer.

The Maximum number of fragments (SAR) setting was set to 2. Segmentation and Re-assembly (SAR) is the process of breaking a packet into smaller units before transmission and re-assembling them into the proper order at the receiving end. Packets are made smaller to speed them through the network, but also due to packet size restrictions in a network. The size of a packet is determined by the smallest maximum PDU in any of the networks in the path. SAR is performed in the transport layer at both ends. The maximum value allowed for this setting is 7. As we matched the MQe packet size to the MNI MTU, a SAR of 2 was optimal.

The Transmit window size is the number of packets that will be put into the network without receiving a TCP-Lite acknowledgement. The Transmit window size is defined in seconds. This was increased to the maximum value of 15, which improved the packet/byte count reduction.

With TCP-Lite enabled, the message flow is illustrated in Figure 13-15.



WMQe Application
MNC
TCP Lite
WECM Mobility Client Stack
IP Stack
Wireline Ethernet Adapter
GPRS WAN Ethernet Adapter
802.1 1b LAN Adapter
WECM Mobility Client Adapter
IP
UDP
WLP
Application Payload

Figure 13-15 TCP-Lite message flow

13.1.11 Setting definitions

From the Gatekeeper, there are a number of settings that may be modified at different levels, for example, an MNC and a Connection Profile both have a `TCP retransmit time to live` setting and a Connection Profile and a TCP-Lite service both have a `Fragment time to live` setting. Where this is the case, the

TCP-Lite value overrides the Connection Profile value, and the Connection Profile value overrides the MNC value.

The following settings should be considered:

- **TCP retransmit time to live**. Configured in the MNC Connection Profile. The time in milliseconds that the TCP-Optimization engine will block retransmitted packets from being delivered. This gives the network provider a chance to deliver the original packet before dealing with the retransmitted packet as well (0 disables this optimization feature).
- **Retransmit interval**. In TCP-Lite resource. The time in milliseconds that the TCP-Lite transport will wait to retransmit a data segment when it has not received an acknowledgement.
- **Sequenced packet time to live**. In MNC resource. The time in seconds before an incomplete packet is discarded – this differs from `Fragment time to live` in that it only applies to Connection Manager v4.1.
- **Fragment time to live**. In connection profile TCP-Lite. The time in seconds before an incomplete packet is discarded.
- **Balance size of PDU fragments**. In connection profile TCP-Lite. In networks which charge per data, it is better to uncheck this to enable full packets to be transmitted rather than to balance the size of packets.
- **Maximum TCP window size**. In connection profile. A TCP sliding window provides flow control by requiring an acknowledgement for a block of data before another block of data is transmitted. The Maximum TCP window size is the maximum number of bytes that can be transmitted before the window opens or closes (if zero, the size of the window is not limited).
- **Minimum TCP window size**. In connection profile. The minimum number of bytes that can be transmitted before the TCP sliding window opens or closes (if zero, the size of the window is not limited).
- **Transmit window size**. In TCP-Lite. Specifies the number of packets that can be sent before waiting for an acknowledgement.
- **Maximum number of fragments (SAR)**. In TCP-Lite. The max number of fragments into which a packet can be segmented, which is used to calculate the max size of a TCP PDU. It is multiplied by the MTU size for the MNC.
- **Transmission delay between fragments**. In connection profile TCP-Lite. The amount of time in milliseconds that is the delay inserted between packet fragment segments
- **Outbound transmission delay**. In connection profile TCP-Lite. The amount of time (ms) that the transmission of packets from the MNC to the client is delayed. The delay lets Connection Manager put multiple packets together and eliminates unnecessary acknowledgements – the default is 100.

- **Outbound ack delay**. In TCP-Lite. The amount of time (ms) that outbound acknowledgements are delayed from being sent – the default is 500.
- **Packet burst rate**. In connection profile. The number of TCP packets that are transmitted before Connection Manager waits for an acknowledgement.
- **Compression algorithm**. In connection profile. The type of compression required for Mobility Clients connecting to the gateway. Options are: PKWare, Zlib, Variant of V.42 BIS, Optional, Mandatory, and Never. Optional is the default and means that no compression is required, and that the Mobility Client can negotiate any compression type.
- **Protocol header reduction**. In connection profile. The options are Mandatory, Never, or Optional. Optional is the default. Mandatory means that header reduction is always performed; Never means that header reduction is not performed. Optional means that the Mobility Client negotiates if header reduction should be performed.
- **TCP protocol optimization**. In connection profile. Specifies whether TCP optimization is attempted for Mobility Clients connecting using this profile. Options are Never, Dynamic, and Fixed. For Dynamic, the Connection Manager monitors the round-trip time and makes adjustments as to how much optimization is done.
- **Maximum number of retransmits**. In TCP-Lite. The maximum number of times a packet can be retransmitted before being discarded.
- **Maximum number of processing threads**. In connection profile. The maximum number of threads used to process transactions. For example, one thread should be allocated for every 25 clients that log in or log out per second.
- **Maximum size of a multi-packet buffer**. In connection profile. The maximum number of bytes of data that can be transmitted as a single chunk. The range is 512 to 3000 bytes.
- **Minimum free space required to load packets**. In connection profile. The minimum number of bytes of available space that should exist in a buffer for more data to be added to it. The range is 4 to 99 bytes.

13.1.12 MQe sample application

A MQe application was developed so we could measure the number of bytes and packets transmitted when sending MQe messages over Connection Manager.

In order to provide consistent results four files of test messages were used for the message payload. The data contained in these files was not duplicate data, but did have a consistent format in that it was of a tabular nature; as a side affect the data contained some white space, which becomes relevant when talking about compression. The four files were as follows:

- 50 messages, sent individually from the server to the client
- 50 messages, sent individually from the client to the server
- 1114 messages, sent in batches of 200 from the server to the client
- 10264 messages, sent in batches of 200 from the client to the server

There are a number of design considerations and configured values that may be used in order to minimize the number of bytes MQe sends across a network. A side effect of these design considerations may also be an optimization for speed.

Messages

As described in the *WebSphere MQ Everyplace Application Programming Guide*, a MQe message payload is held in one or more MQeFields objects; these are containers with a type, name, and value. MQeFields may be recursive so it is possible for an application developer to create messages containing multiple MQeFields objects depending upon the type of data required. The MQeFields object is self describing allowing MQe to dispense with a static header object to describe the data payload. Therefore, the amount of MQe information added to a message is partly dependent upon the number of MQeFields objects in the MQeMsgObject. The best approach to minimize network usage is to put all the data into one MQeMsgObject with the smallest name appropriate for the application. This has two effects, firstly the amount of MQe data in the message payload is minimized, and secondly the time taken to serialize and de-serialize the message is also minimized. The down side of this approach is that your application becomes responsible for parsing the data in the message rather than being able to use multiple MQeFields objects.

Every time data is sent across the network, additional bytes are added by the network protocol; for instance, TCP adds 20 bytes, then IP adds a further 20 bytes. The application developer can minimize this overhead by creating fewer large messages rather than sending numerous small messages. Also see “Adapter” on page 272.

If your MQe messages are destined for a WMQ queue, then you may now use a MQe API that allows you to put multiple messages into a single object, MQeMultiMsgObject. This object allows the application developer to wrap multiple messages into one large message with the corresponding support for retrieving the messages. MQeMultiMsgObject removes the responsibility of parsing a large message for imbedded smaller messages from the application. The MQeMultiMsgObject was not applicable for the tests undertaken as part of the work on which this paper is based.

Attributes

Attributes are set either on a particular queue or on a specific message. An MQeAttribute may contain one of each of the following in any combination: MQeCompressor, MQeCryptor, MQeAuthenticator. We will discuss the compressor and cryptor as they have an affect upon the number of bytes flowed across the network.

Compressor

MQe allows the application developer to select the best compression algorithm for the data being used by the application. In general, the compressor is best selected from empirical testing using data that will be generated by the application in production. For instance, in our tests we found that when sending the file of 50 messages, both from client to server and visa versa, we found the MQeRleCompressor class was marginally better than the MQeGZIPCompressor. However, when sending the large messages created by combining 200 messages, the MQeGZIPCompressor achieved a considerably better compression, roughly halving the number of packets and reducing the number of bytes by approximately one third over the RLE compressor.

It should be noted that Connection Manager has its own compressor, so even if no compressor is configured in MQe, with the correct configuration in Connection Manager it is possible to see a 50% reduction in the number of bytes flowed across the network. Using a MQe compressor suitable to the data, it is possible to attain a five-times compression of data.

Cryptor

Typically, when encrypting data, the amount of data in bytes is increased; this is no exception with MQe. When using Connection Manager the data is automatically encrypted across the network and does not appear to dramatically increase the number of bytes. However, if using a MQe cryptor we saw results ranging from very little difference to a doubling of the number of bytes depending upon the type of data. The important thing to note is if encryption is required before the data is put to the network. If only network encryption is required then using the Connection Manager encryption is optimal, however, if encryption is required between the MQe API and the network and to the permanent storage, then a MQe cryptor will need to be used.

13.1.13 Queue and queue manager names

The names of the queues and queue managers are sent across the network as part of the MQe information added to the message payload. The names for these resources should be kept as short as possible.

13.1.14 Communications

There are a number of objects instantiated as part of MQe communications, which although not exposed to the application developer they may be configured to help optimize network usage. Many of these configuration values may be set using administration messages, the details of which may be found in the *MQe Configuration Guide*. Figure 13-16 shows the MQe communications objects.

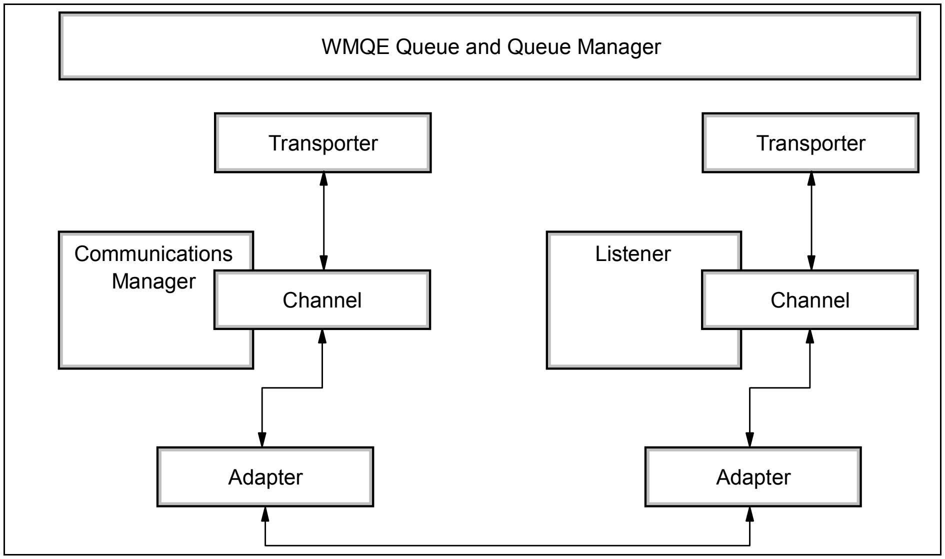


Figure 13-16 MQe communications objects

Transporter (not exposed to the API)

The MQeTransporter class is responsible for establishing a link from one queue to another; there is a one-to-one relationship between an MQeQueue and an MQeTransporter. The MQeTransporter is not configured in any way, and is included for completeness.

Channel (not exposed to the API)

The MQeChannel class is responsible for establishing a link between queue managers. As can be seen from the diagram, the object holding a reference to an open channel alters, depending on whether the channel is incoming or outgoing. This difference results in a different location for setting the channel time-out value, which is of interest to us when optimizing the topology for network usage.

The channel time-out value is given in milliseconds and is used to close channels that have not been used for a given period of time. A background thread is responsible for closing the channels if they have been idle for the period of time set by the channel time-out value. It is important to try to make the time-out values in the client and server as close as possible. The reason for this is that TCP/IP will put a socket into FINWAIT2 state when a socket is closed. If the corresponding socket on the other end of the connection does not respond, then this state can be held either indefinitely or until the operating system times it out. The channel time-out value needs to be long enough for a client and server to have finished sending and receiving messages, but not so long that channels take up valuable system resources. The default value is 5 minutes.

For outgoing channels the channel time-out is set on the queue manager, for incoming channels it is set in the listener. It is worth noting that when a listener is initially configured and started, the underlying thread responsible for the channel time-out does not know the listener exists, and therefore does not call the channel time-out until another channel time-out expires. For testing purposes, it therefore may be worthwhile creating the listener, setting the channel time-out, then stopping and restarting the queue manager.

Adapter

The communications adapter provides the protocol support for sending the data across the network. There are a number of communications adapters shipped with MQe; it is also possible for the application developer to write their own adapter should they so wish. The communications adapters have a number of configured values, the details of which are provided in the *Application Reference Manual*. We will look at those values that have an impact on network utilization:

Packet size

The packet size has the most profound affect upon the number of packets sent across the network. We found the best way of setting this is to do a series of tests using a network sniffer tool.

An Ethernet LAN typically will have an maximum transmission unit (MTU) of 1500, however, this may be lowered by a router. First of all, set the packet size to whatever you believe the MTU to be, allowing for the TCP and IP headers; you may also have to allow for the network, for instance, Ethernet puts 14 bytes information at the head of each packet. In this instance therefore, we would probably set the packet size to 1446. Now send messages from one MQe queue manager to another of a size greater than 1446 so that the adapter has to split the data up. By looking at the results from the sniffer tool you should see some IP fragmentation that will determine the MTU of the network.

When using MQe with Connection Manager it is possible to take one of two approaches. Either match the packet size of MQe with Connection Manager, in which case you will need to do some more testing to find out what the header size is like being used by Connection Manager, which may vary. Or, set the packet size in MQe to be as large or larger than the largest message you are likely to send. In this case we will leave the splitting of the data entirely to

Connection Manager. It is probably more effective to use the latter approach as this removes the necessity to understand the header construction of Connection Manager.

We found our MTU to be 1429, allowing for the maximum possible header used by Connection Manager; we set the packet size to 1349.

Time-out

The adapter will attempt to read from a socket until the time-out value is exhausted. Internally, a static low value is used to allow the adapter to ascertain if the thread on which the adapter is running has been stopped providing timely response to closing the queue manager. The time-out value should be commensurate with the speed of the network.

13.2 Glossary

GPRS = General Packet Radio System

IP = Internet Protocol

MNC = Mobile Network Connection

MNI = Mobile Network Interface

MTU = Maximum Transmission Unit

MSS = Maximum Segment Size

PDA = Personal Digital Assistant

PDU = Protocol Data Unit

TCP = Transmission Control Protocol

UDP = User Datagram Protocol

VPN = Virtual Private Network

WECM = WebSphere Everyplace Connection Manager

WLP = Wireless Link Protocol

MQe = WebSphere MQ Everyplace

14

HTTP Access Services

Today, more and more users access information and applications using the HTTP protocol such as company Web pages, applications delivered through Web services, and client server applications that use HTTP tunneling. In this chapter we present an overview of the HTTP Access Services as implemented in WebSphere Connection Manager. A sample scenario is also included to illustrate the required configuration.

14.1 Overview

HTTP Access Services in Everyplace Connection Manager allow network external users to securely access internal resources that use HTTP transport layer providing a secure tunnel for HTTP access using Secure Socket Layer (SSL).

It is not unusual for a company to have internal servers not designed to be secure, allowing anonymous access without data encryption to anyone on the enterprise intranet. For these servers HTTP Access Services can add access control (authentication), confidentiality (encryption), and authorization (certification).

Even for servers deployed with SSL, there are several benefits in having HTTP Access Service. The most relevant benefits are:

- Servers can be out of the DMZ (only the Everyplace Connection Manager box needs to be located in the DMZ).
- There is no need to open ports through the external firewall.
- All external SSL connections are to Everyplace Connection Manager. Clients only need to recognize and trust the Everyplace Connection Manager digital certificate. This allows deploying a widely recognized digital certificate from a well known authority (for example, Verisign, RSA, and Thawte) on the Everyplace Connection Manager machine.

Internal servers can use less expensive certificates or even self-signed certificates since they only need to be trusted by resources within the enterprise.

This topic has some architectural approaches, describes a typical HTTP Access Service Scenario, and shows what you need to do to have it up and running.

Note: The HTTP Access Service in Connection Manager does not require the use of the Mobility Client software in the client device. That is, the Wireless Link Protocol (WLP) is not used in this case.

Many unique architectures can be built to deliver HTTP access using Everyplace Connection Manager HTTP Access Services. In this chapter the following scenarios are covered:

- Multiple Web application servers behind a reverse proxy server, using a single host name
- Multiple Web application servers behind a reverse proxy, using multiple host names

14.1.1 Multiple Web application servers using a single host name

Often the requirements include access to multiple Web application servers. The previous configuration can be extended by adding a reverse proxy server between Everyplace Connection Manager and the Web application servers as shown in Figure 14-1.

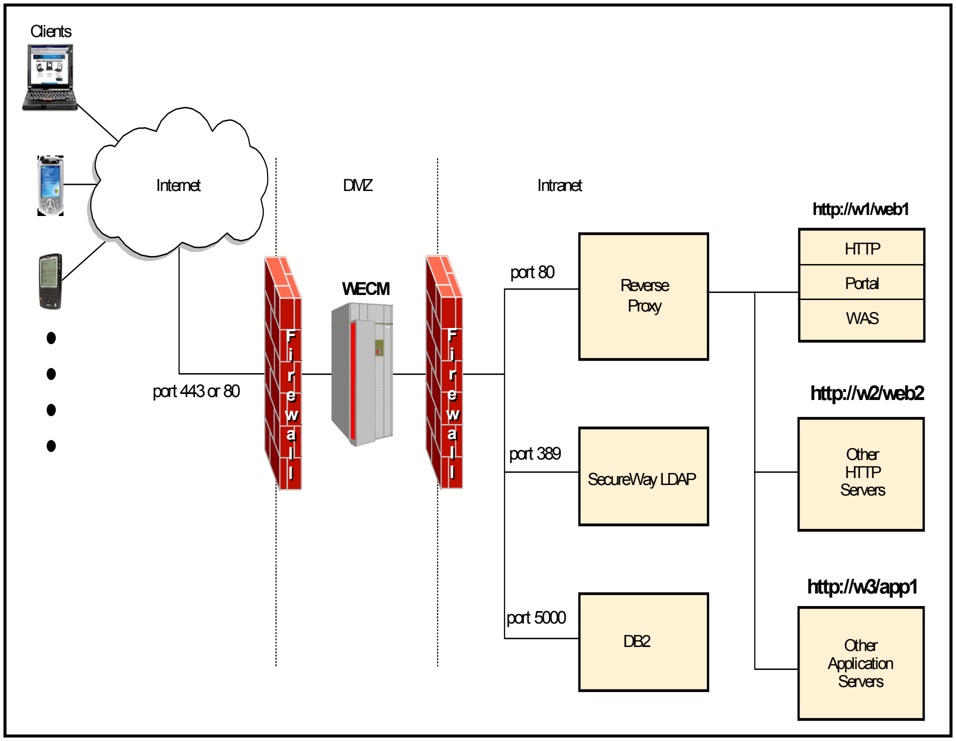


Figure 14-1 Multiple Web application servers using a single host name

In this case a reverse proxy such as IBM WebSphere Edge Server includes a reverse proxy component and passes HTTP traffic to one of several servers based on the URL received from the Everyplace Connection Manager machine.

This configuration has several advantages. For example:

- Secure access to multiple Web application servers
- All Web sites applications are exposed through a single host name, giving a corporate site a consistent look.
- Servers can easily be added, deleted, turned off, etc. (requires no change on the clients URL).
- A single digital certificate is used.

14.1.2 Multiple Web application servers using multiple host names

In some cases the URL naming scheme used in 14.1.1, “Multiple Web application servers using a single host name” on page 277 may not always work. For example, an enterprise may not want to organize Web pages and applications on existing servers in order to support the URL naming conventions required. Another problem is that servers may have different URLs when accessed from the intranet than through Everyplace Connection Manager.

Many reverse proxy servers allow translation of the hostname, not just the URI. In this case it is possible to use the configuration illustrated in Figure 14-2.

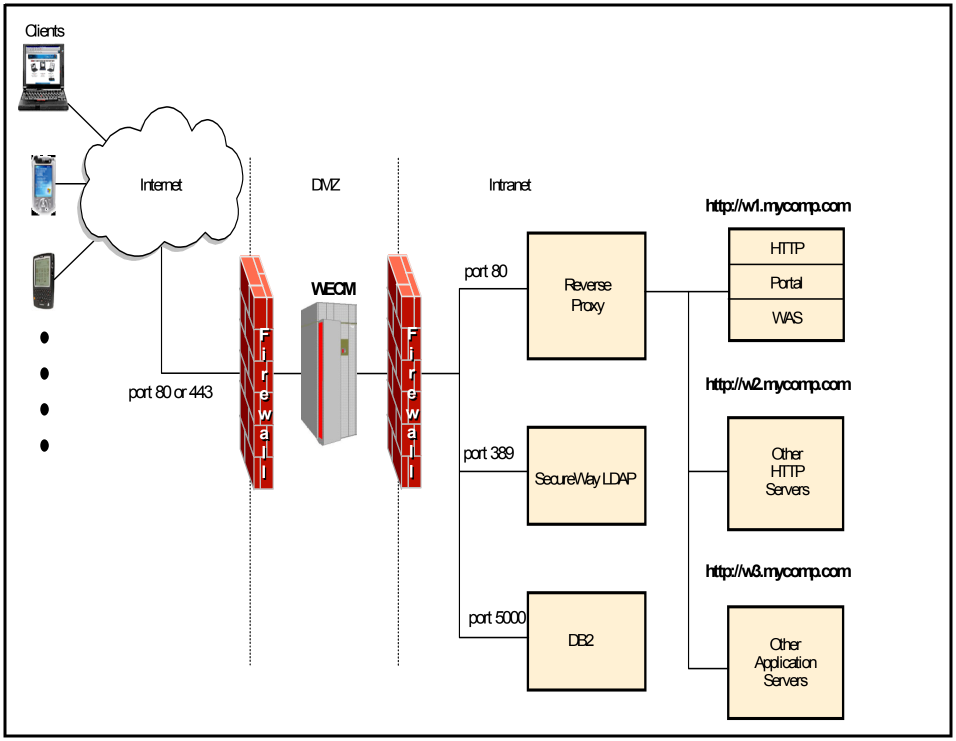


Figure 14-2 Multiple Web application servers using multiple host names

With this configuration secure access can be allowed to any Web application server in the mycompany.com domain. To allow access to a `http://wx.mycompany.com` (x can be in that scenario 1, 2, or 3) using this configuration requires the following:

- Everyplace Connection Manager needs a valid IP address on the Internet.
- The internal server’s host names must resolve to the IP address of the Everyplace Connection Manager machine using a public DNS.

- An internal DNS server is used by the reverse proxy.
- The reverse proxy must be configured to proxy host names with `w1.mycompany.com` translating to `w1.mycompany.com`.
- Everyplace Connection Manager machine must have a valid wildcard digital certificate for `*.mycompany.com`.

This scheme works because the host name in the URL `http://w1.mycompany.com` is resolved differently by clients on the Internet and the proxy server using the internal DNS. This results in the client sending HTTP requests to the Everyplace Connection Manager machine, which the proxy then forwards to the correct internal IP address of the actual server. Significantly, this scheme uses the same server host name internally and externally avoiding the need for the complex URL rewriting that would otherwise be required.

The wildcard certificate is required because the Everyplace Connection Manager can only have one certificate. It will use this certificate to create a secure SSL regardless of the URL used to reach the Everyplace Connection Manager machine. If all servers are in the `mycompany.com` domain and the digital certificate is used to `*.mycompany.com` by a certificate authority recognized by the user will not see any warnings. If the servers are not in the same domain or a wildcard certificate is not used, the connection can still be made but the user will be warned that the server name in the certificate does not match the server in the URL entered by the user.

Note: Secure Sockets Layer (SSL) is required between HTTP clients and Connection Manager. HTTP requests using configured unsecure ports (default is port 80) will be rerouted to the HTTPS secure port (default 443).

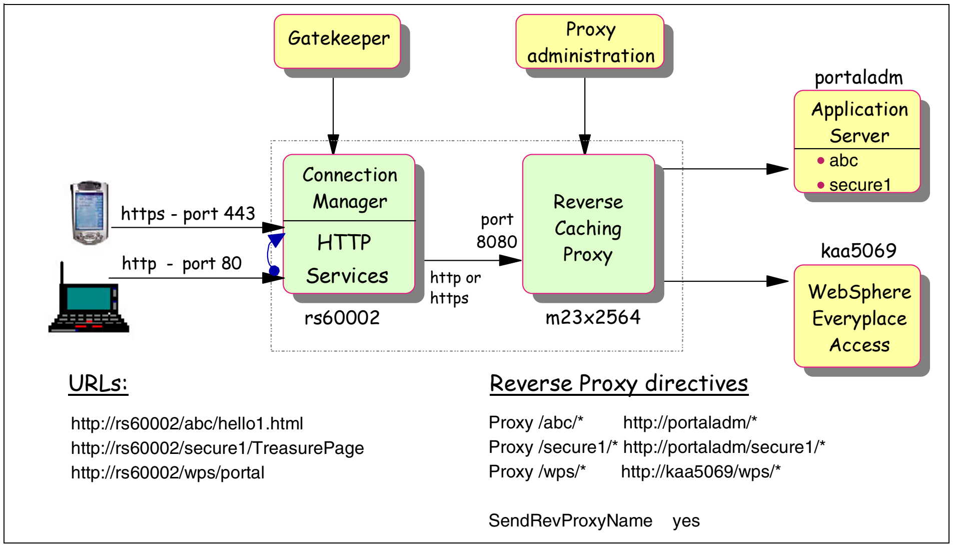
14.2 Sample scenario

In this section, a sample scenario shown in Figure 14-3 is configured to illustrate the HTTP Access Services function implemented in Connection Manager. In this sample scenario, the following seven machines are used:

1. Everyplace Connection Manager machine
   - IBM AIX version V5.0.0.2
   - IBM DB2 Universal Database
   - IBM Directory Services V4.1
   - IBM WebSphere Connection Manager V5.0
2. Reverse proxy server
   - Microsoft Windows 2000 Server
   - IBM WebSphere Edge Server V2.0

3. Application server 1
   - Microsoft Windows 2000 Server
   - IBM WebSphere Application Server V5.0
4. Application server 2
   - Microsoft Windows 2000 Server
   - IBM WebSphere Everyplace Access V4.3
5. Gatekeeper client
   - Microsoft Windows 2000 Professional
   - IBM WebSphere Connection Manager Gatekeeper V5.0
6. HTTP client
   - Microsoft Windows 2000 Professional
   - Microsoft Internet Explorer V6.0
7. PDA Client
   - Microsoft PocketPC 2002

The sample scenario is illustrated in Figure 14-3.



URLs:

http://rs60002/abc/hello1.html
http://rs60002/secure1/TreasurePage
http://rs60002/wps/portal

Reverse Proxy directives

```
Proxy /abc/*       http://portaladm/*
Proxy /secure1/* http://portaladm/secure1/*
Proxy /wps/*       http://kaa5069/wps/*

SendRevProxyName    yes
```

Figure 14-3 HTTP Access Services sample scenario

Application server 1

Application server 1 runs two applications. The first application is called Hello1, which is basically a static HTML page. The second application is called TreasurePage, which is cataloged inside WebSphere Application Server.

TreasurePage is a protected servlet application and requires login credentials (user ID and password) for HTTP basic authentication.

Application server 2

Application server 2 runs WebSphere Everyplace Access (WEA). This scenario uses the portal component in WebSphere Everyplace Access. For more information about Everyplace Connection Manager and WebSphere Everyplace Access integration, see Chapter 15, “WebSphere Everyplace Access integration” on page 303.

Reverse proxy server

The reverse proxy machine plays a major role in the architecture; it is responsible to redirect the Everyplace Connection Manager messages to the application servers and vice-versa.

The reverse proxy must be specifically configured to work with Everyplace Connection Manager. Configuration options need to be made in the ibmproxy.conf file. For Windows machines this file is located in the C:\Program Files\IBM\edge\cp\etc\en_US directory, assuming that you are using default values when you installed IBM WebSphere Edge Server.

Port directive

The port used by the reverse proxy must be updated (for example, port 8080) to match the HTTP Access Service configuration. The HTTP configuration is shown in Example 14-1.

Example 14-1 Port directive

```
#       Port directive:
#       Port used by the server.
#       Default:  80
#       Syntax:   Port <num>
Port 8080
```

SendRevProxyName directive

This directive must be configured so that application servers will also send HTTP traffic back to the reverse proxy. The configuration is illustrated in Example 14-2.

Example 14-2 SendRevProxyName

```
#       SendRevProxyName directive:
#
#       In a reverse proxy scenario, WTE normally sends the destination
#       origin server name in the HOST header of the request to the origin
#       server.  If this directive is set to yes, WTE will instead send
#       the WTE host name in the HOST header of the request to the origin
#        server.  This allows the origin server to use the WTE host name in
#        redirects sent back.  Therefore, subsequent requests to redirected
#        locations will go through WTE.
#
#        Default:  no
#        Syntax:   SendRevProxyName <yes | no>
#
# Example:
# SendRevProxyName      yes

SendRevProxyName  yes
```

Mapping rules

The mapping rules need to be updated as they vary for different scenarios; this is because applications use different paths for their content location. The added directives for this sample scenario are illustrated in Example 14-3; also see Figure 14-3.

Example 14-3 Mapping rules

```
# ====================================================================== #
#        Mapping rules
# ====================================================================== #
Pass  /Docs/htmldocs/* C:\PROGRA~1\IBM\edge\doc\en_US\*
Pass  /Docs/* C:\PROGRA~1\IBM\edge\cp\server_root\Docs\en_US\*
Pass  /httpd-internal-icons/* C:\PROGRA~1\IBM\edge\cp\server_root\cpicons\*
Pass  /icons/* C:\PROGRA~1\IBM\edge\cp\server_root\cpicons\*
Pass  /wsApplet/* C:\PROGRA~1\IBM\edge\cp\server_root\admin-bin\webexec\*
Pass  /Admin/*.gif C:\PROGRA~1\IBM\edge\cp\server_root\Admin\en_US\*.gif
Pass  /Admin/*.html C:\PROGRA~1\IBM\edge\cp\server_root\Admin\en_US\*.html
#
# HTTP Access Services changes - sample scenario
# Pass  /* C:\PROGRA~1\IBM\edge\cp\server_root\pub\en_US\*
Pass  /pub/*     C:\PROGRA~1\IBM\edge\cp\server_root\pub\en_US\*

Proxy /wps/*http://9.24.104.224/wps/*
Proxy /abc/*http://9.24.104.19/*
Proxy /secure1/*http://9.24.104.19/secure1/*
# End of HTTP Access Services changes#
```

14.2.1 Creating a Connection Manager server certificate

HTTP Access Services in Everyplace Connection Manager requires secure connections with client devices by using Secure Sockets Layer (SSL). This section describes the IBM AIX Key Management utility tool configuration required to create the server certificate to establish a secure connection with HTTP clients.

The connection between Everyplace Connection Manager and the reverse proxy can also be secure with SSL; however, this is optional and therefore it is not documented in this chapter.

Note: A self-signed certificate is used for this scenario; however, for better security it is recommended that you obtain an import a server certificate from an authorized Certificate Authority (CA).

For example, follow these steps:

1. Log on to Everyplace Connection Manager with the right privileges.
2. Open a terminal window and enter `ikeyman`
3. The IBM Key Management window should come up. Then select **Key Database File -> Open** as shown in Figure 14-4.

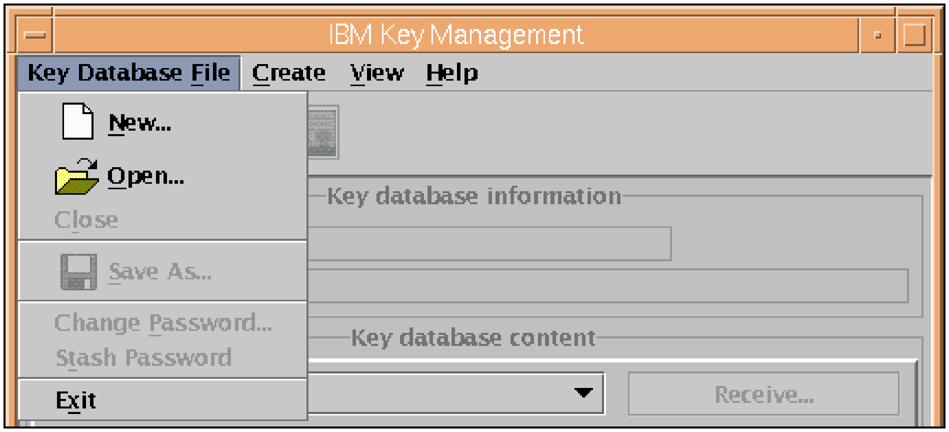
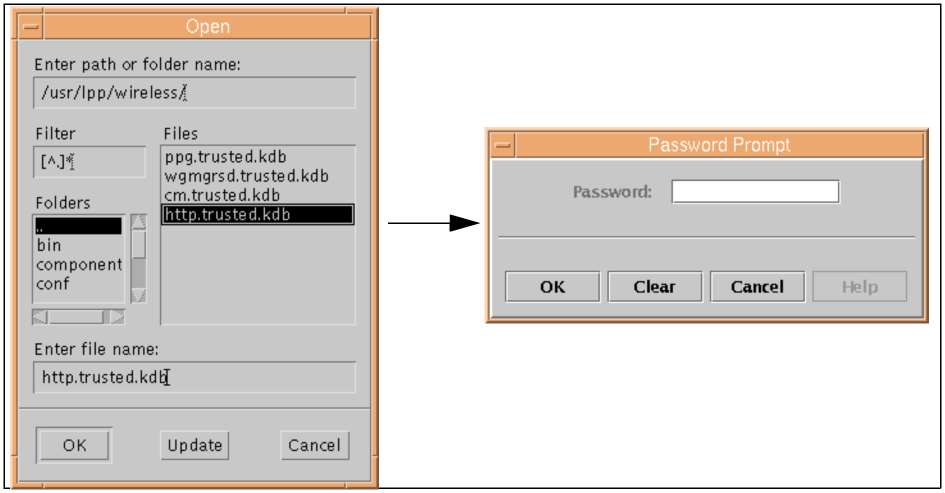


Figure 14-4 IBM Key Management initial window

4. Input the database path according to your system standards. For example, in this scenario the default key database values are used. Therefore, the file name for the key database is /usr/lpp/wireless/http.trusted.kdb as shown in Figure 14-5.
5. Click **OK** and to enter the password.



Enter path or folder name: /usr/lpp/wireless/

Filter: [^.]*

Files:
ppg.trusted.kdb
wgmgrsd.trusted.kdb
cm.trusted.kdb
http.trusted.kdb

Folders:
..
bin
component
conf

Enter file name: http.trusted.kdb

Password:

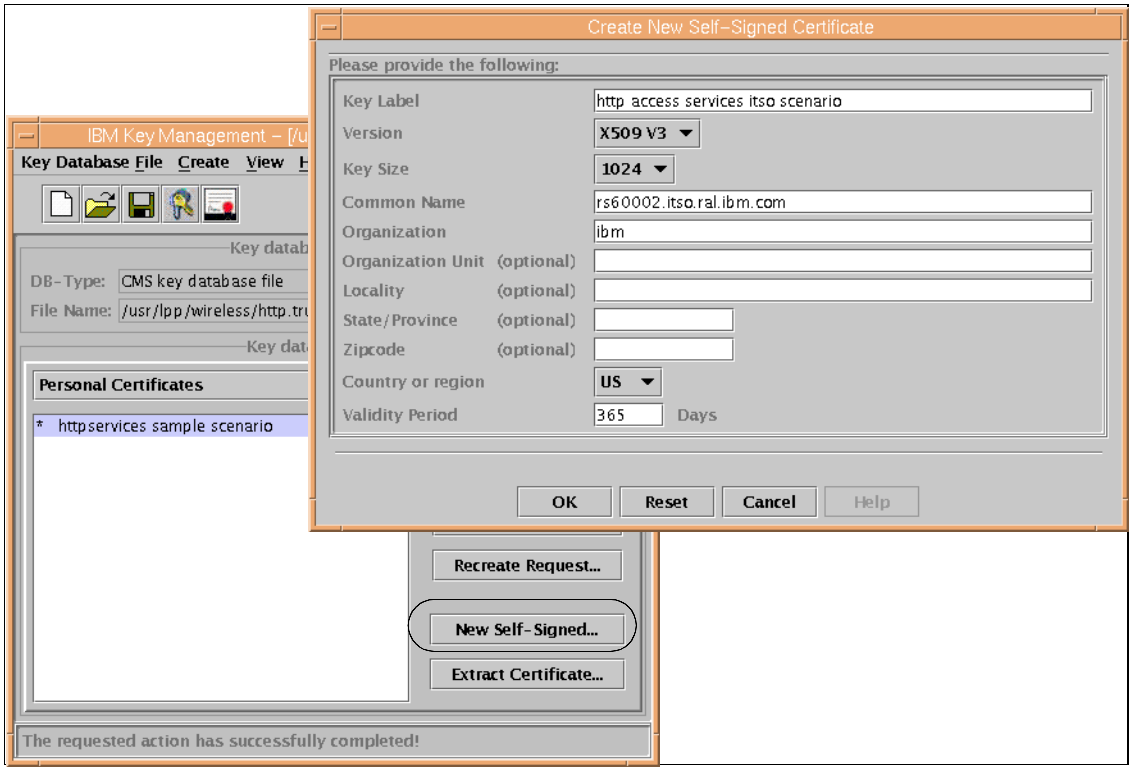
Figure 14-5 Open Key Database

6. The default password used by Everyplace Connection Manager is `trusted`. Initially, enter this password and click **OK** to access the key database as illustrated in Figure 14-6.

   Note: In this window there is already a personal certificate. This should not be the case the first time you access IBM Key Management.

7. For this sample scenario a self-signed certificate is used. Click the **New Self-Signed** button. The Create New Self-Signed Certificate form appears.

8. Provide the information needed to create the certificate; for an example, see Figure 14-6.



Create New Self-Signed Certificate

Please provide the following:

Key Label: http access services itso scenario

Version: X509 V3

Key Size: 1024

Common Name: rs60002.itso.ral.ibm.com

Organization: ibm

Organization Unit (optional):

Locality (optional):

State/Province (optional):

Zipcode (optional):

Country or region: US

Validity Period: 365 Days

Key datab

DB-Type: CMS key database file

File Name: /usr/lpp/wireless/http.tru

Key dat

Personal Certificates

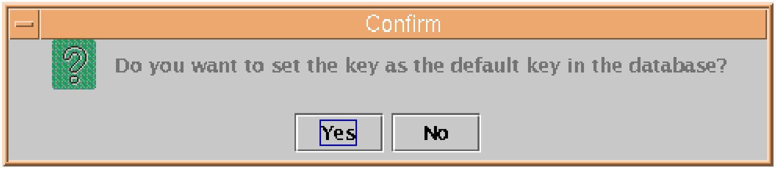
* httpservices sample scenario

The requested action has successfully completed!

Figure 14-6 Create a New Self-Signed certificate

9. Click **OK** to create the certificate. If there is more than one certificate created, you will be asked if the certificate will be the default (active) key in the database. Select **Yes** to confirm as shown in Figure 14-7.

   Note: You can store many certificates in the database but only one certificate can be active at the server side.



Confirm

Do you want to set the key as the default key in the database?

Figure 14-7 Set the default key

10. Exit the IBM Key Management tool once the self-signed certificate has been created.

14.2.2 HTTP Access Services configuration

In this section a HTTP Access Services sample configuration is included. The Gatekeeper administration tool is used for this purpose. For example:

1. Log in to the Gatekeeper, right-click and add HTTP Access Service as illustrated in Figure 14-8.

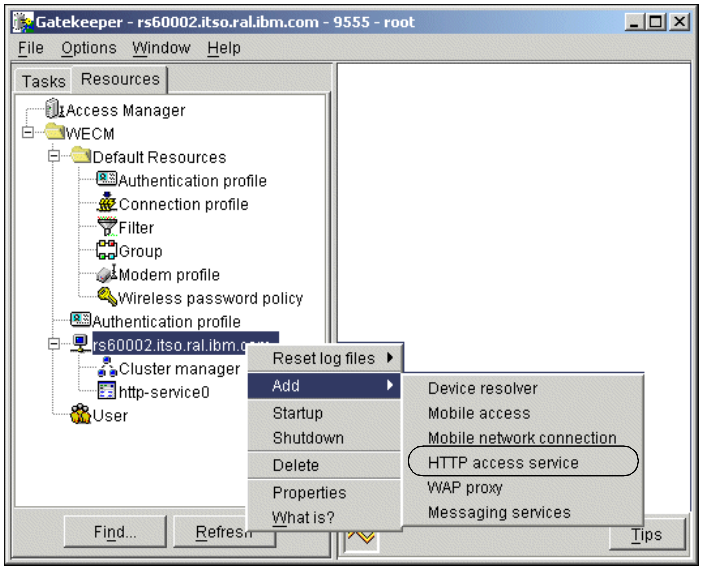
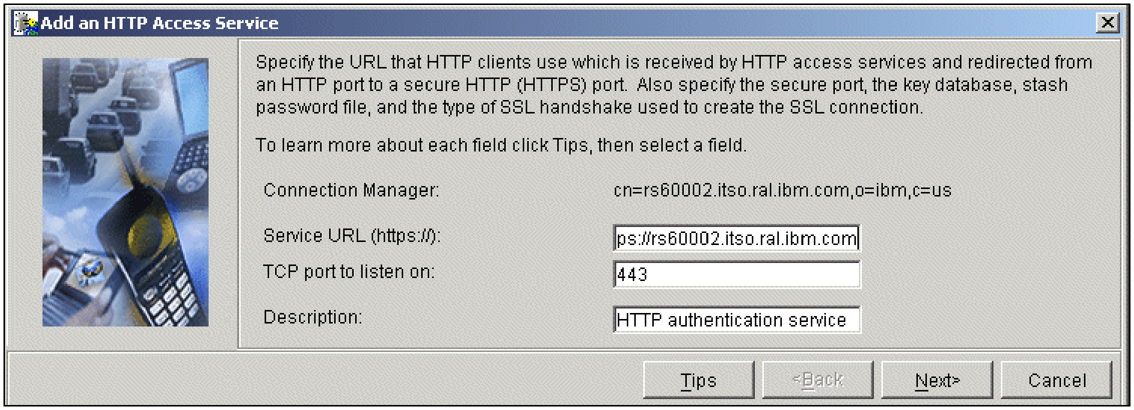


Figure 14-8 Add HTTP Access Service

2. Specify the service URL used by the HTTP clients and the secure port.



Specify the URL that HTTP clients use which is received by HTTP access services and redirected from an HTTP port to a secure HTTP (HTTPS) port. Also specify the secure port, the key database, stash password file, and the type of SSL handshake used to create the SSL connection.

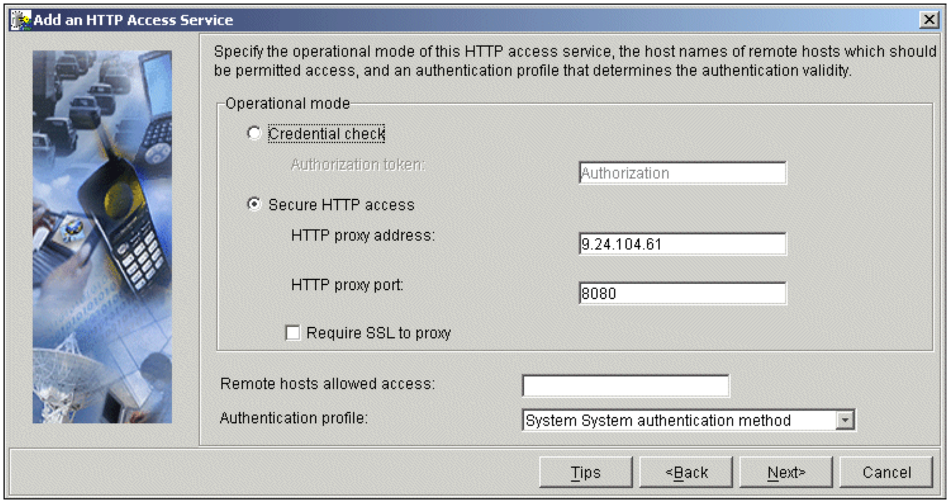
To learn more about each field click Tips, then select a field.

Connection Manager:	cn=rs60002.itso.ral.ibm.com,o=ibm,c=us
Service URL (https://):	ps://rs60002.itso.ral.ibm.com
TCP port to listen on:	443
Description:	HTTP authentication service

Figure 14-9 Adding HTTP Access Services

Note: Unsecure client requests will be redirected to the secure port.

3. Configure the operational mode used by HTTP Access Service. In this sample scenario secure HTTP access is used. The HTTP proxy address is the IP address of the reverse proxy machine as illustrated in Figure 14-3 on page 280, and 8080 is the listening port.



Specify the operational mode of this HTTP access service, the host names of remote hosts which should be permitted access, and an authentication profile that determines the authentication validity.

Operational mode

○ Credential check

Authorization token: Authorization

● Secure HTTP access

HTTP proxy address: 9.24.104.61

HTTP proxy port: 8080

□ Require SSL to proxy

Remote hosts allowed access:

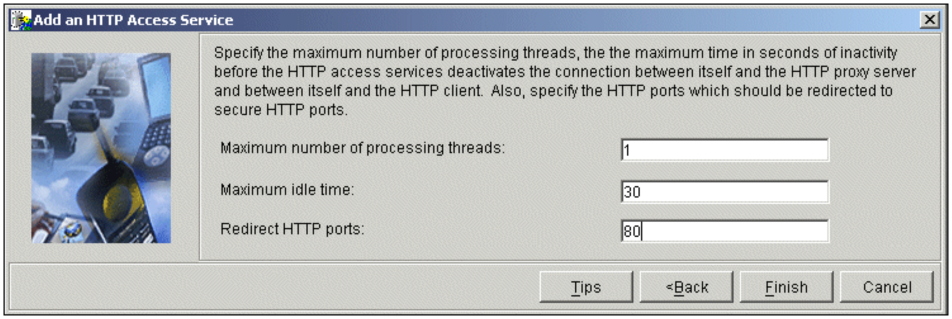
Authentication profile: System System authentication method

Figure 14-10 Reverse proxy configuration

4. Select the **Require SSL to proxy** box if HTTP Access Services will also use a secure connection to the reverse proxy machine (optional).

   Note: As an alternative, HTTP Access Services can also receive client credentials from an application server in the HTTP header. If this option is used, HTTP Access Services will check the validity of the credentials according to the authentication profile selected (the default is Authorization). However, this option does not support Lightweight Third Party Authentication (LTPA) for Single Sign-On (SSO). For information about SSO with LTPA tokens, see Chapter 17, “WebSphere Everyplace Access Single Sign-On using LTPA” on page 347.

5. List the HTTP ports that will be redirected to the secure port. In this sample scenario, only the unsecure port 80 is used. Take the default values or provide the maximum number of processing threads and the maximum idle time if needed.



Add an HTTP Access Service

Specify the maximum number of processing threads, the the maximum time in seconds of inactivity before the HTTP access services deactivates the connection between itself and the HTTP proxy server and between itself and the HTTP client. Also, specify the HTTP ports which should be redirected to secure HTTP ports.

Maximum number of processing threads: 1

Maximum idle time: 30

Redirect HTTP ports: 80

Figure 14-11 Configure unsecure ports to be redirected

Note: It is important to guarantee that these ports are not used by any other application in the Connection Manager machine. For example, an HTTP server required to access LDAP Directory Services should be closed or its configuration changed to use different ports, otherwise, port conflicts will occur.

14.3 Running the sample scenario

Before running the scenario, you may want to recycle Everyplace Connection Manager. This procedure guarantees that all modifications done are considered.

For example, follow these steps:

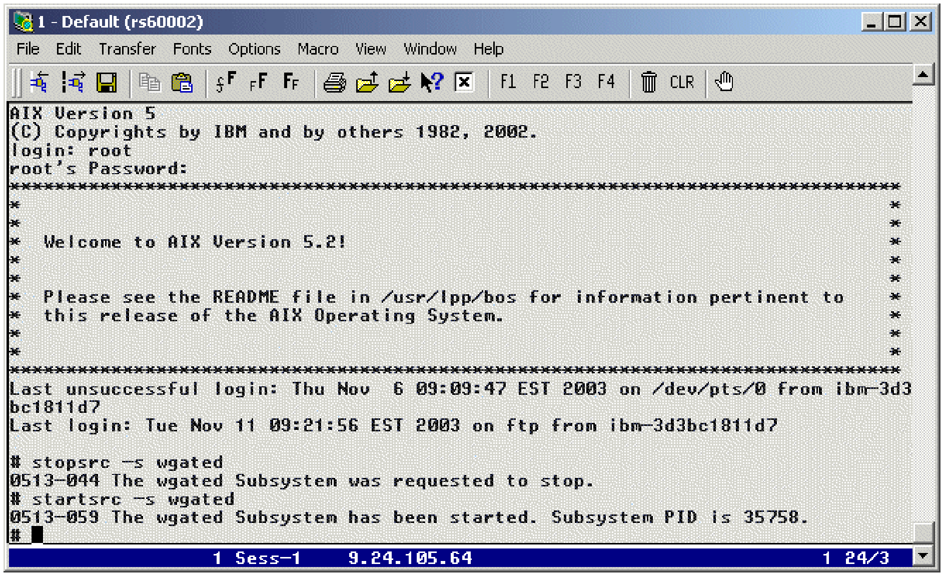
1. Log in to the Everyplace Connection Manager box and enter the following command to stop the services:

   ```
   stopsrc -s wgated
   ```

2. Wait for approximately 30 seconds to make sure Connection Manager has stopped and enter the following command to re-start the services:

   ```
   startsrc -s wgated
   ```

3. Wait for approximately 30 seconds to make sure Everyplace Connection Manager has started. Figure 14-12 shows a successful refresh.



```
AIX Version 5
(C) Copyrights by IBM and by others 1982, 2002.
login: root
root's Password:
*******************************************************************************
*                                                                             *
*                                                                             *
*  Welcome to AIX Version 5.2!                                                *
*                                                                             *
*                                                                             *
*  Please see the README file in /usr/lpp/bos for information pertinent to    *
*  this release of the AIX Operating System.                                  *
*                                                                             *
*                                                                             *
*******************************************************************************
Last unsuccessful login: Thu Nov  6 09:09:47 EST 2003 on /dev/pts/0 from ibm-3d3
bc1811d7
Last login: Tue Nov 11 09:21:56 EST 2003 on ftp from ibm-3d3bc1811d7

# stopsrc -s wgated
0513-044 The wgated Subsystem was requested to stop.
# startsrc -s wgated
0513-059 The wgated Subsystem has been started. Subsystem PID is 35758.
#
```

Figure 14-12 Everyplace Connection Manager Refresh procedure

The sample scenario included in this chapter has been tried using the following client devices connected to Everyplace Connection Manager:

- Desktop browser client running on a Windows machine
- PDA client, a Pocket PC is used

The following sample applications are executed:

- Hello1 application. A Web application providing static content.
- Treasure Page Application. A Web application (servlet) requiring credentials (user ID and password) for HTTP Basic Authentication
- Portal Application. A Everyplace Access portal application using several portlet applications and supporting desktop browser and PDAs.

14.3.1 Desktop clients

In this section a desktop browser is used (Internet Explorer) to access the sample applications. The access is through Connection Manager and its associated reverse proxy.

Hello1 application

The hello1 sample application is accessed from a desktop using the IE browser. For example:

1. Open Internet Explorer and enter the following URL:

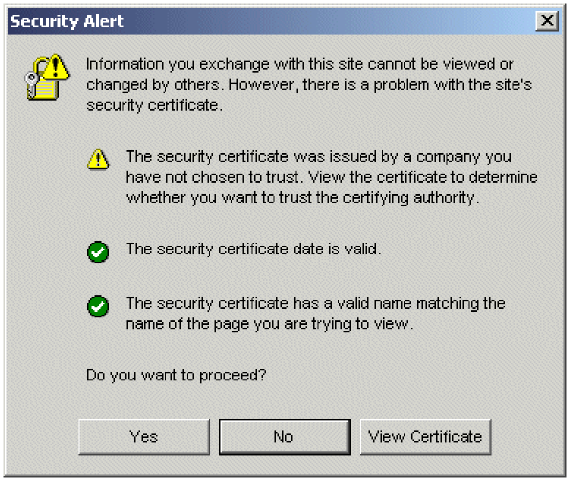
   `https://rs60002.itso.ral.ibm.com/abc/hello1.html`

   Note: If HTTP is used, the request will be redirected to the secure port, assuming that the unsecure port (default 80) is also configured for redirection as illustrated in Figure 14-11 on page 288.

   The proxy directive for application *abc* makes the reverse proxy redirect the request to the proper application server (see Example 14-3 on page 282).

2. In this scenario a self-signed certificate is used for the secure connection. Therefore, you will receive a security alert because the certificate signer cannot be found and cannot be trusted. In a real situation you probably would like to use a certificate signed by the known Certificate Authority (CA).

   The alert is shown in Figure 14-13. For testing purposes, click **Yes** to accept the certificate.



Security Alert

Information you exchange with this site cannot be viewed or changed by others. However, there is a problem with the site's security certificate.

⚠ The security certificate was issued by a company you have not chosen to trust. View the certificate to determine whether you want to trust the certifying authority.

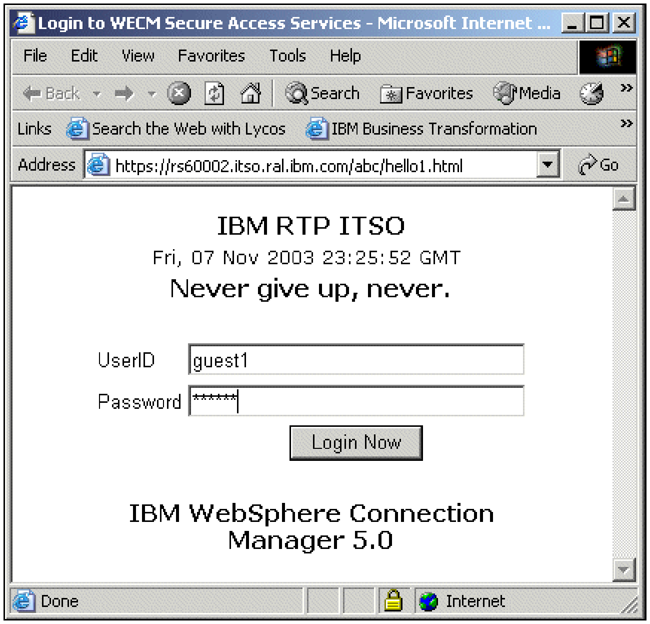
✓ The security certificate date is valid.

✓ The security certificate has a valid name matching the name of the page you are trying to view.

Do you want to proceed?

Figure 14-13 Security alert using self-signed certificate

3. The first time you connect, Everyplace Connection Manager will prompt you with the login window. Enter the user ID and password previously created and click **Login Now** as shown in Figure 14-14.



Login to WECM Secure Access Services - Microsoft Internet ...

https://rs60002.itso.ral.ibm.com/abc/hello1.html

IBM RTP ITSO

Fri, 07 Nov 2003 23:25:52 GMT

Never give up, never.

UserID guest1

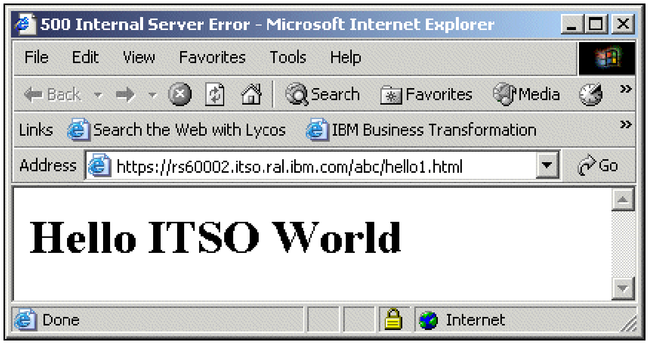
Password ******

Login Now

IBM WebSphere Connection Manager 5.0

Figure 14-14 Everyplace Connection Manager login

4. If you entered the correct user ID and password, you will be authenticated by Connection Manager, and the application will run as shown in Figure 14-15.



500 Internal Server Error - Microsoft Internet Explorer

https://rs60002.itso.ral.ibm.com/abc/hello1.html

Hello ITSO World

Figure 14-15 Desktop browser HTTP Services application access

Treasure Page application

The Treasure Page sample application is a protected application and requires HTTP Basic Authentication. In this scenario, it is accessed from a desktop browser (Internet Explorer) as illustrated in Figure 14-17.

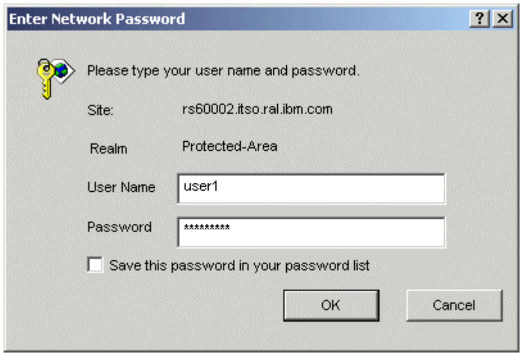
For example, follow these steps:

1. The following URL is used to access this application through Everyplace Connection Manager and the configured reverse proxy:

   ```
   https://rs60002.itso.ral.ibm.com/secure1/TreasurePage
   ```

   Note: See Example 14-3 on page 282 for details on the configuration of the reverse proxy directive for this URL (secure1).

2. The first time you connect to Connection Manager, you will be prompted for proper credentials (see Figure 14-14).
3. In addition, because this application requires HTTP basic authentication, you will also be prompted for application credentials as shown in Figure 14-16.



Enter Network Password

Please type your user name and password.

Site: rs60002.itso.ral.ibm.com

Realm Protected-Area

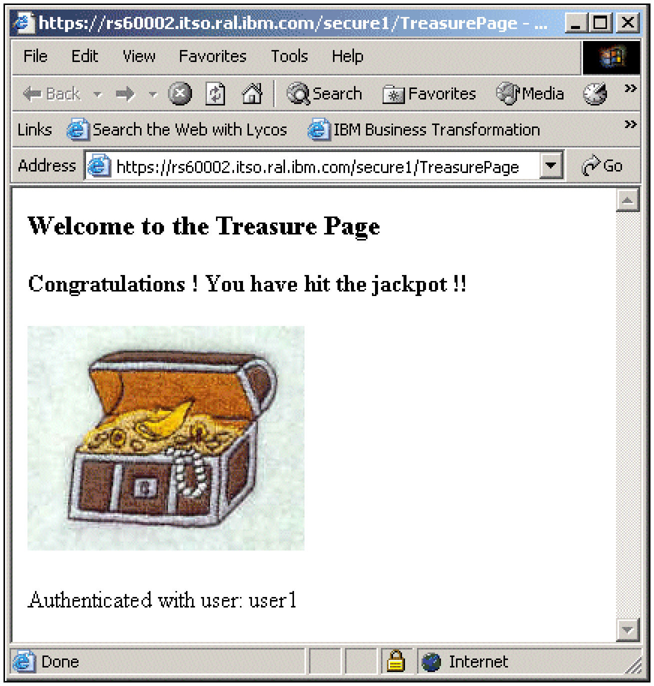
User Name user1

Password *********

□ Save this password in your password list

Figure 14-16 TreasurePage login window (HTTP basic authentication)

4. Once you are authenticated, the application executes and sends the response to the reverse proxy, Connection Manager, and to the desktop browser as illustrated in Figure 14-17.


https://rs60002.itso.ral.ibm.com/secure1/TreasurePage - ...
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Search the Web with Lycos IBM Business Transformation
Address https://rs60002.itso.ral.ibm.com/secure1/TreasurePage Go
Done Internet

Welcome to the Treasure Page

Congratulations ! You have hit the jackpot !!

Authenticated with user: user1

Figure 14-17 TreasurePage application results

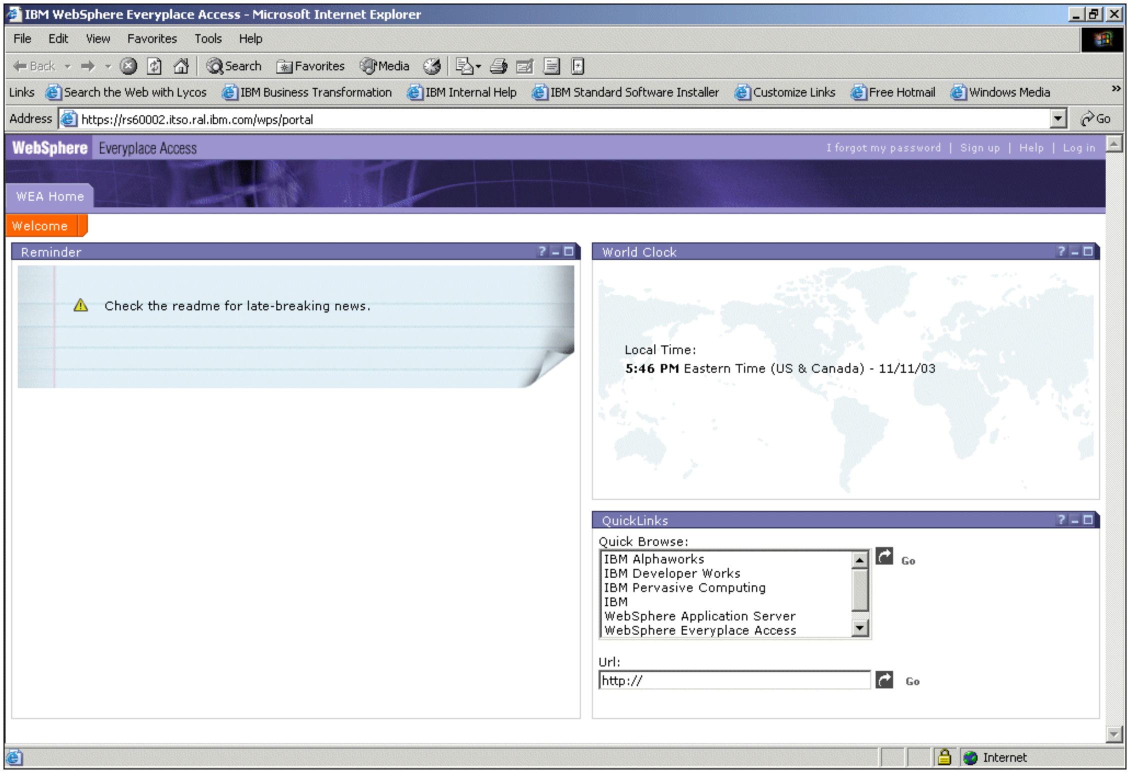
Portal application

The portal sample application is a set of WebSphere Everyplace Access portlet applications. For example, follow these steps to access the portal:

1. Open Internet Explorer and enter the following URL:

   ```
   https://rs60002.itso.ral.ibm.com/wps/portal
   ```

2. If required, click **Yes** to accept the self-signed certificate for the secure connection (SSL) with Connection Manager.
3. If required, enter the proper user ID and password to log in to Connection Manager and click **Login Now** as shown in Figure 14-14 on page 291.
4. If the user ID and password were correct, the Portal welcome page should come up as shown in Figure 14-18.


IBM WebSphere Everyplace Access - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Search the Web with Lycos IBM Business Transformation IBM Internal Help IBM Standard Software Installer Customize Links Free Hotmail Windows Media
Address
Go
WebSphere Everyplace Access
I forgot my password | Sign up | Help | Log in
WEA Home
Welcome
Reminder
World Clock
QuickLinks
Quick Browse:
IBM Alphaworks
IBM Developer Works
IBM Pervasive Computing
IBM
WebSphere Application Server
WebSphere Everyplace Access
Go
Url:
Go
Internet

Address: https://rs60002.itso.ral.ibm.com/wps/portal

Check the readme for late-breaking news.

Local Time:
5:46 PM Eastern Time (US & Canada) - 11/11/03

Url: http://

Figure 14-18 Portal application access using Connection Manager

14.3.2 PDA clients

In this section a PDA (Pocket PC) is used to access the sample applications. The access is through Connection Manager and its associated reverse proxy.

Hello1 application

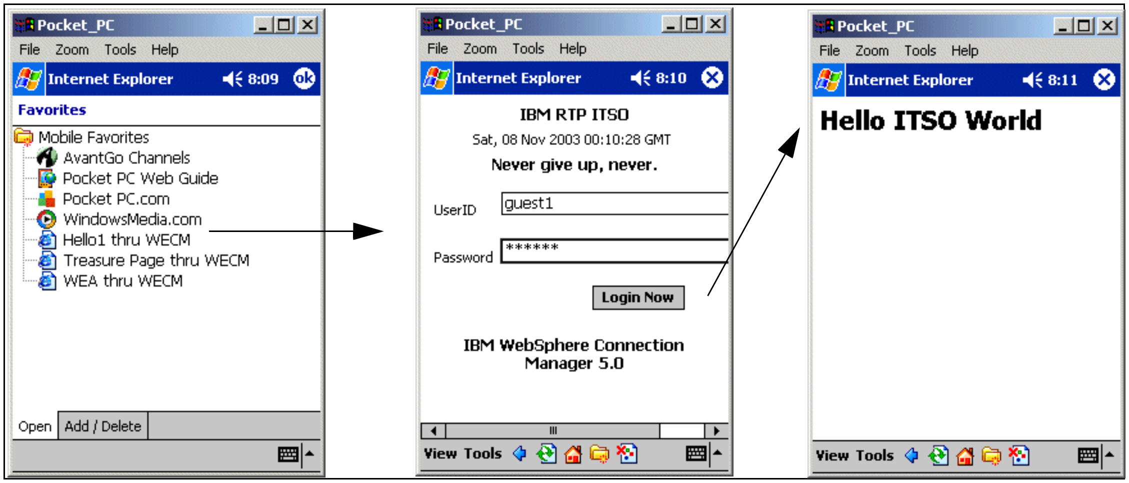
The hello1 sample application is accessed from a Pocket PC using the IE browser as illustrated in Figure 14-19. For example:

1. Open Internet Explorer.
2. Click **Favorites**. You should see the URL applications added to the Mobile Favorites folder.
3. Select the **hello1 application**. It uses the following URL:

```
https://rs60002.itso.ral.ibm.com/abc/hello1.html
```

Note: See Example 14-3 on page 282 for details on the configuration of the reverse proxy directive for this URL (abc).

4. The Security Window appears. Click **Yes**.
5. The first time you connect, Everyplace Connection Manager will prompt you with the login window. Enter the user ID and password previously created.
6. Click **Login Now**.
7. The application is accessed and the results are shown.



IBM RTP ITSO

Sat, 08 Nov 2003 00:10:28 GMT

Never give up, never.

UserID guest1

Password ******

IBM WebSphere Connection Manager 5.0

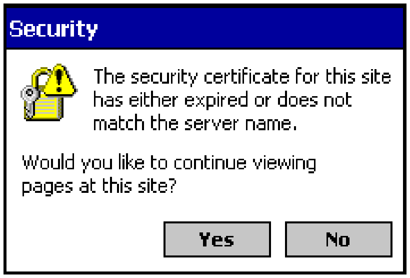
Hello ITSO World

Figure 14-19 Accessing the hello1 application from Pocket PC

If the server certificate cannot be validated during the SSL connection to Everyplace Connection Manager, you will see a security window asking you if you want to proceed with the request.

Note: This situation happens also when the server uses a self-signed certificate since no Certificate Authority (CA) has signed the server certificate. That is, it is signed by the same owner (server), which may be unknown to the browser.

The security window is shown in Figure 14-20, and for testing purposes click **Yes** to accept the certificate and continue with the request.



Security

The security certificate for this site has either expired or does not match the server name.

Would you like to continue viewing pages at this site?

Figure 14-20 Browser in Pocket PC cannot validate server certificate

Treasure Page application

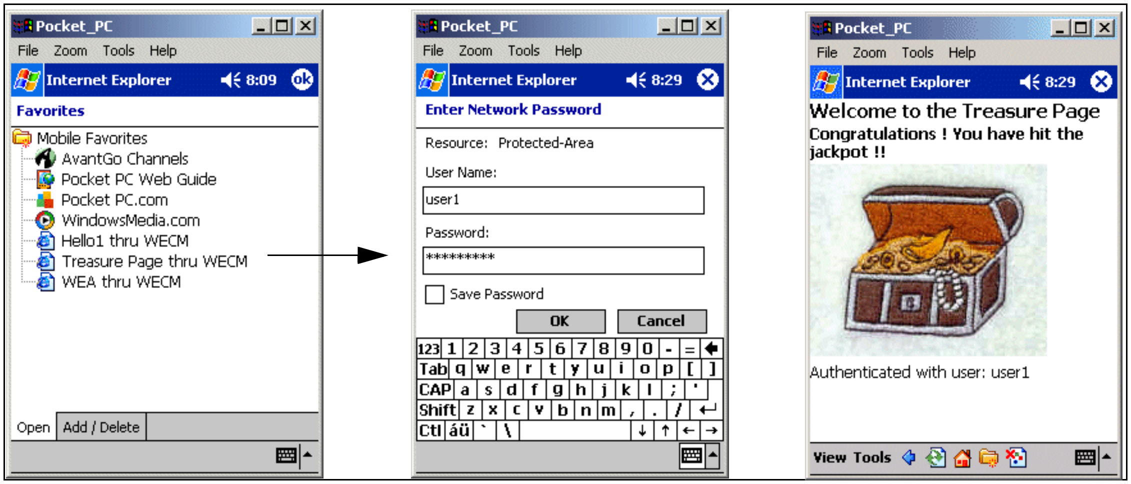
The Treasure Page sample application is a protected application and requires HTTP basic authentication. In this scenario, it is accessed from a Pocket PC using the IE browser as illustrated in Figure 14-21.

The following URL is used to access this application through Everyplace Connection Manager and the configured reverse proxy:

```
https://rs60002.itso.ral.ibm.com/secure1/TreasurePage
```

Note: See Example 14-3 on page 282 for details on the configuration of the reverse proxy directive for this URL (secure1).

For example, select the application in the Pocket PC Internet Explorer Favorites.



Enter Network Password

Resource: Protected-Area

User Name:

user1

Password:

****

Welcome to the Treasure Page

Congratulations ! You have hit the jackpot !!

Authenticated with user: user1

Figure 14-21 TreasurePage application access from Pocket PC

Since this application is protected and requires authentication (HTTP basic authentication), you will be required to log in twice as follows:

1. The first time you connect to Everyplace Connection Manager, you will be prompted by Connection Manager to enter a valid user ID and password; see Figure 14-19 for details.
2. When you access the application requiring basic authentication you will be prompted by the Pocket PC browser to enter the credentials (user ID and password) as illustrated in Figure 14-21.

WebSphere Everyplace Access portal application

The portal application is a set of WebSphere Everyplace Access portlet applications. For example, follow these steps to access this server:

1. Open Internet Explorer in the Pocket PC.
2. Click **Favorites**. You should see the URL applications moved to the Mobile Favorites folder.

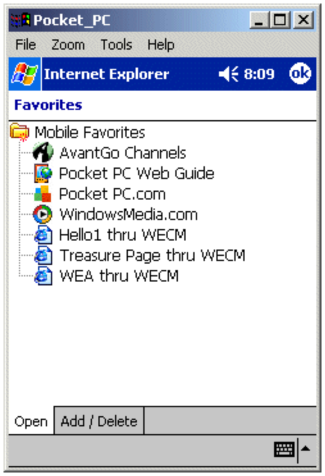


Figure 14-22 Portal application access

3. Click **WEA thru WECM.** The following URL is used in this case:

   `https://rs60002.itso.ral.ibm.com/wps/portal`

4. The first time you connect, Everyplace Connection Manager prompts you with a login window, enter the proper user ID and password (see Figure 14-19 on page 295) and click **Login Now** to access the Portal welcome page customized for PDAs.

5. The Portal welcome page is illustrated in Figure 14-23 for this sample application. Notice that for PDAs portlets are initially in a minimized state and therefore there will be an icon for each portlet.

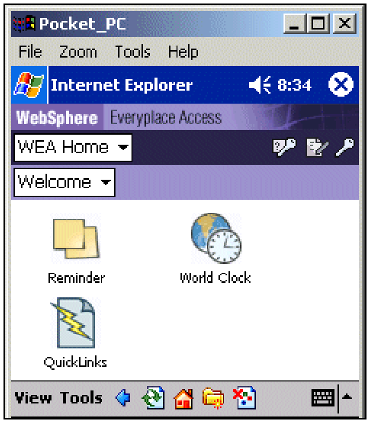


Figure 14-23 TreasurePage application access results

6. The next stepis to log in as a user to WebSphere Everyplace Access. For details see Chapter 15, "WebSphere Everyplace Access integration" on page 303.

14.4 Troubleshooting

This section provides some guidelines to help you troubleshoot potential problems you may encounter when implementing HTTP Access Services in Everyplace Connection Manager.

Application problems

In general, you will need to make sure that the applications run properly when not going through Everyplace Connection Manager. For example, for the sample scenario included in this chapter, the following URLs can be used to access the applications directly from a desktop browser:

- `http://portaladm/hello1.html`
- `http://portaladm/secure1/TreasurePage`
- `http://kaa5069.itso.ral.ibm.com/wps/portal`

Reverse proxy problems

It is recommended that you follow the request progress using the reverse proxy activity monitor function. For example, follow these steps:

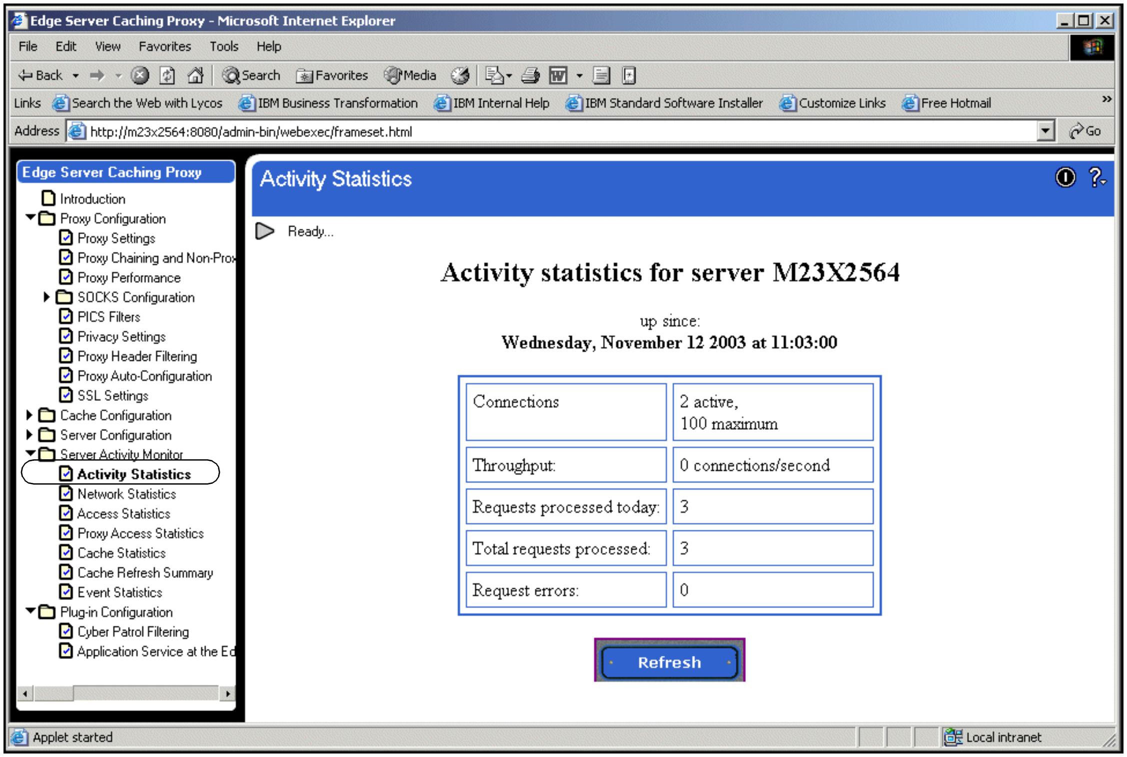
1. Access the WebSphere Edge Server Caching Proxy front page. Remember that the port directive has been changed in the configuration, and you will need to use the proper value. In the sample scenario included in this chapter port 8080 is used. Therefore, to invoke the WebSphere Edge Server Caching Proxy front page the following URL should be used:

   ```
   http://m23x2564:8080/pub/FrntPage.html
   ```

2. Click **Configuration and Administration Forms** and log on to caching proxy administration.
3. Click **Server Activity Monitor -> Activity Statistics**.

Note: It is a good procedure to clear the activity statistics before running your scenario by refreshing the IBM Caching Proxy service.

Figure 14-24 illustrates the statistics for the reverse proxy traffic.



Address: http://m23x2564:8080/admin-bin/webexec/frameset.html

Activity statistics for server M23X2564

up since:
Wednesday, November 12 2003 at 11:03:00

Connections	2 active, 100 maximum
Throughput:	0 connections/second
Requests processed today:	3
Total requests processed:	3
Request errors:	0

Figure 14-24 IBM WebSphere Edge Server activity statistics

HTTP Access Services startup problems

To make sure HTTP Access Services has been started properly, open the wg.log file. This file is located on /var/adm and look for HTTP_Service. Table 14-4 shows a port 80 conflict when starting the services.

Example 14-4 Unsuccessful HTTP Access Services start up (port 80 conflict)

```
31566:  258 (Nov 12 03/11:10:45.5652): HTTP_Service::startup (entry)
31566:  258 (Nov 12 03/11:10:45.5708): TcpPort::listen (entry)
31566:  258 (Nov 12 03/11:10:45.5710): TcpPort::open (entry)
31566:  258 (Nov 12 03/11:10:45.5719): TcpPort::open (return), rc=0
31566:  258 (Nov 12 03/11:10:45.5722): TcpPort::listen (return), rc=0
31566:  258 (Nov 12 03/11:10:45.5725): http-service0: connection established
(TCP/listen:443)
31566:  258 (Nov 12 03/11:10:45.5727): TcpPort::listen (entry)
31566:  258 (Nov 12 03/11:10:45.5727): TcpPort::open (entry)
31566:  258 (Nov 12 03/11:10:45.5729): TcpPort::open (return), rc=0
31566:  258 (Nov 12 03/11:10:45.5774): TcpPort: unable to bind socket, port 80
(Address already in use)
31566:  258 (Nov 12 03/11:10:45.5776): TcpPort::listen (return), rc=67
31566:  258 (Nov 12 03/11:10:45.5778): http-service0: failed to establish
connection, (TCP/listen:80) (Address already in use)
31566:  258 (Nov 12 03/11:10:45.5779): TcpPort: closing fd '11'
```

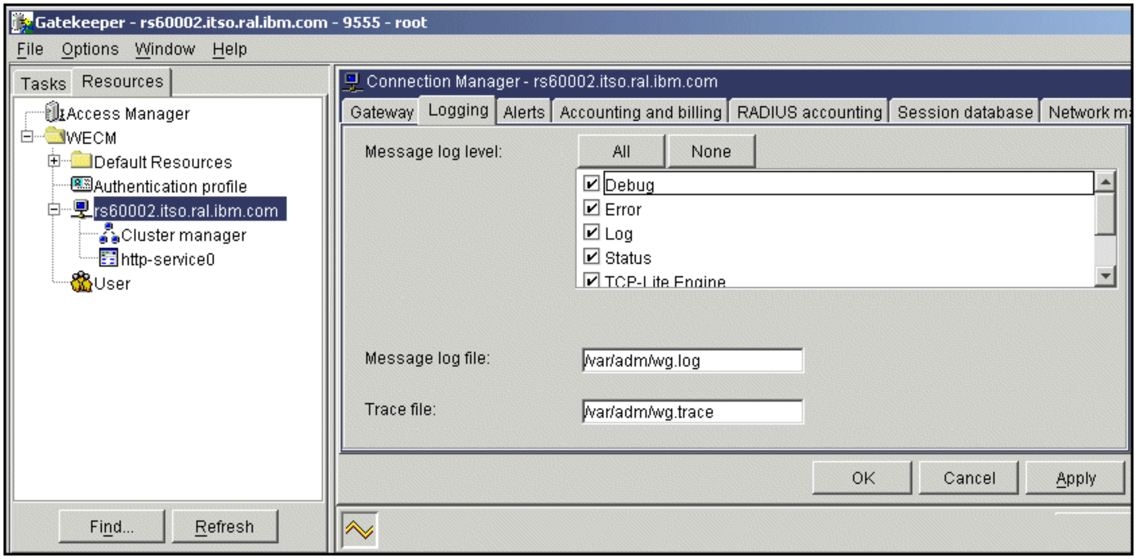
In this case, you will need to find the process using port 80 and resolve the conflict. As a reference, Table 14-5 shows a successful HTTP Access Service startup.

Example 14-5 Successful HTTP Access Services start up

```
10102:  258 (Nov 12 03/11:22:37.6527): HTTP_Service::startup (entry)
 10102:  258 (Nov 12 03/11:22:37.6582): TcpPort::listen (entry)
 10102:  258 (Nov 12 03/11:22:37.6586): TcpPort::open (entry)
 10102:  258 (Nov 12 03/11:22:37.6589): TcpPort::open (return), rc=0
 10102:  258 (Nov 12 03/11:22:37.6591): TcpPort::listen (return), rc=0
 10102:  258 (Nov 12 03/11:22:37.6594): http-service0: connection established
(TCP/listen:443)
 10102:  258 (Nov 12 03/11:22:37.6596): TcpPort::listen (entry)
 10102:  258 (Nov 12 03/11:22:37.6597): TcpPort::open (entry)
 10102:  258 (Nov 12 03/11:22:37.6599): TcpPort::open (return), rc=0
 10102:  258 (Nov 12 03/11:22:37.6601): TcpPort::listen (return), rc=0
 10102:  258 (Nov 12 03/11:22:37.6602): http-service0: connection established
(TCP/listen:80)
```

Everyplace Connection Manager logging

It is recommended to enable logging when troubleshooting HTTP Access Services problems in Everyplace Connection Manager. A sample configuration is shown in Figure 14-25 where logging levels are set, and the location of the log files are defined.



Message log level:
☑ Debug
☑ Error
☑ Log
☑ Status
☑ TCP-Lite Engine

Message log file: /var/adm/wg.log

Trace file: /var/adm/wg.trace

Figure 14-25 Logging configuration

15

WebSphere Everyplace Access integration

This chapter presents an overview on how WebSphere Everyplace Connection Manager and WebSphere Everyplace Access work together. The following topics are considered in this chapter:

- Typical WebSphere Everyplace Access and Connection Manager configurations
- Why you will need to integrate Connection Manager and WebSphere Everyplace Access
- A typical configuration using Connection Manager and WebSphere Everyplace Access
- LDAP considerations
- Secure remote access using Connection Manager HTTP Access
- How to build a WebSphere Everyplace Access and Connection Manager integrated scenario

15.1 WebSphere Everyplace Access integration

Connection Manager is commonly found in enterprises with a significant number of mobile users. WebSphere Everyplace Access is similarly targeted at mobile users. Its themes, skins, and transcoding technology generate Web content optimized for a wide variety of devices including desktop systems, laptops, PDAs, and WAP browsers. Indeed, IBM has designed WebSphere Everyplace Access and Connection Manager to work together.

WebSphere Everyplace Access is intended to extend corporate applications, e-mail, and PIM data to mobile users. Given the sensitive nature of such data, many enterprises choose to locate WebSphere Everyplace Access servers on their intranets. A Connection Manager server is then used to allow mobile users access to the WebSphere Everyplace Access servers.

Connection Manager provides WebSphere Everyplace Access users two ways to securely access their WebSphere Everyplace Access servers:

- For portal access through a Web browser, Connection Manager provides secure HTTP access.
- Using the Mobility Client, Connection Manager provides full access to the WebSphere Everyplace Access server (portal, synchronizations services, device management, alerts, etc.).

Note: Using either method Connection Manager improves security by providing robust authentication, authorization, and confidentiality (by means of encryption).

15.1.1 Common configurations

Configuring Connection Manager and WebSphere Everyplace Access in a large enterprise can involve a large number of separate servers. A typical configuration may include:

- Connection Manager servers
- WebSphere Everyplace Access portal servers
- WebSphere Everyplace Access synchronization
- Corporate LDAP
- Database servers
- Domino servers
- Application servers
- Sametime® servers
- Intelligent notification servers

> **Note:** Scalability and redundancy requirements can further add complexity to the scenario.

15.1.2 LDAP considerations

Both WebSphere Everyplace Connection Manager and WebSphere Everyplace Access use Lightweight Directory Access Protocol (LDAP) to store and retrieve user and configuration information. In addition, the following options should be taken into consideration when integrating these two products:

- Connection Manager and WebSphere Everyplace Access both offer the choice of creating a local LDAP or using a remote LDAP.
- Connection Manager can also be configured in a split LDAP configuration with configuration data and user data in separate LDAPs.

In general, the LDAP strategy is a complex approach, and it will not be seen in detail in this redbook. However, some highlights are important to understand in a WebSphere Everyplace Access and Connection Manager integration, and this chapter attempts to cover the most important issues in a few common scenarios.

The following configurations can be considered:

Case 1: Using separate user directories

This is the non-integrated approach where Connection Manager and Everyplace Access implement their own LDAP independent LDAP directory. The scenario is illustrated in Figure 15-1.

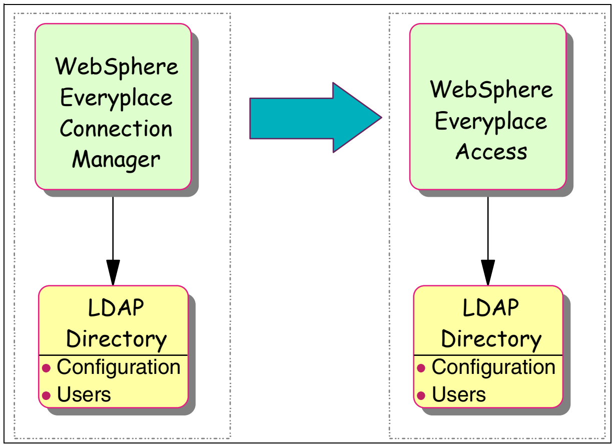

WebSphere Everyplace Connection Manager
LDAP Directory
Configuration
Users
WebSphere Everyplace Access
LDAP Directory
Configuration
Users

Figure 15-1 Separate LDAP directories

This configuration where Connection Manager and WebSphere Everyplace Access use separate LDAP directory servers, and gives you the benefit that LDAP directories run on different and independent machines, which gives you better performance. However, it creates a user management overhead since users must reside on both directories.

Note: This is the sample scenario described in Chapter 14, “HTTP Access Services” on page 275.

Case 2: Sharing user directories

In this scenario, you configure Connection Manager to access the user entries residing in the Everyplace Access LDAP Directory. Only the user directory is shared and the Connection Manager configuration values still reside in its own LDAP directory.

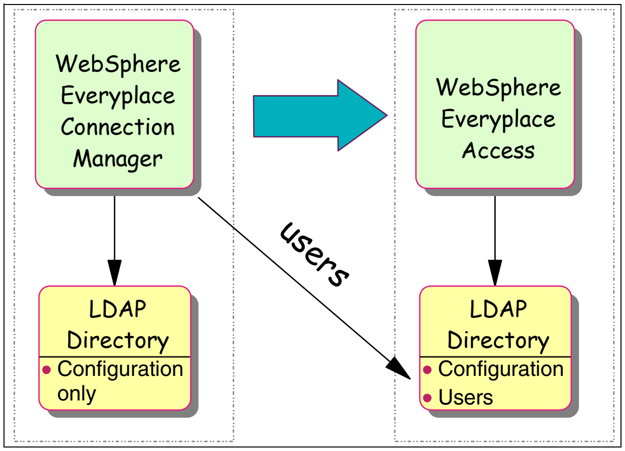


Figure 15-2 Sharing user LDAP Directory

This scenario gives you the following benefits:

- Centralized user management in a single directory
- Deploying the Everyplace Access LDAP Directory behind the firewalls provides better security for the user information.

Note: This is the approach included as a sample scenario in this chapter. See 15.2, “Sample scenario” on page 308.

Case 3: Sharing user directory and configuration

In this scenario, you configure Connection Manager to access both its configuration values and the user entries residing in the Everyplace Access LDAP directory. The scenario is illustrated in Figure 15-3.

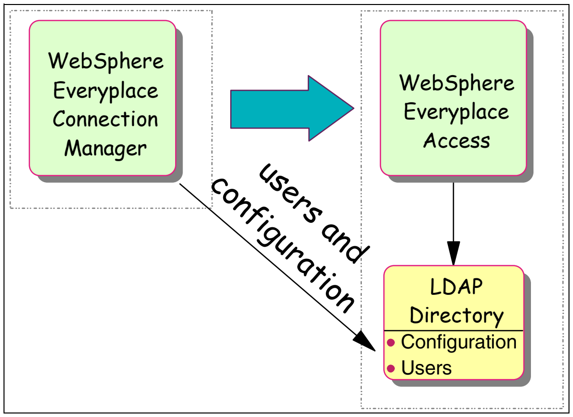

WebSphere Everyplace Connection Manager
WebSphere Everyplace Access
users and configuration
LDAP Directory
Configuration
Users

Figure 15-3 Sharing LDAP for users and configuration

This scenario gives you the following benefits:

- Centralized user management in a single directory
- Deploying the Everyplace Access LDAP Directory behind the firewalls provides better security for both the Connection Manager configuration values and the user information.

The drawback in this scenario is that Connection Manager has a dependency on WebSphere Everyplace Access for configuration changes.

Case 4: LDAP Enterprise Directory

In this scenario, you configure both Connection Manager and WebSphere Everyplace Access to access a centralized LDAP Directory. The scenario is illustrated in Figure 15-4.

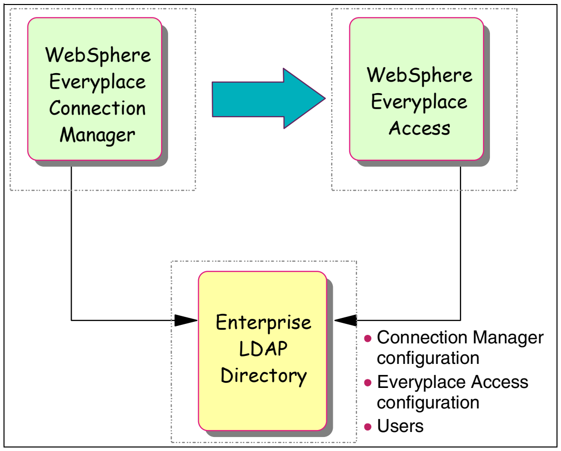

WebSphere Everyplace Connection Manager
WebSphere Everyplace Access
Enterprise LDAP Directory
• Connection Manager configuration
• Everyplace Access configuration
• Users

Figure 15-4 LDAP Enterprise Directory

This scenario gives you the following benefits:

- Centralized user management in a single directory
- The enterprise LDAP directory resides behind the firewalls and therefore provides better security for both the Connection Manager configuration values and the user information.

The drawback in this scenario is that Connection Manager has a dependency on the enterprise LDAP directory for configuration changes.

15.2 Sample scenario

The scenario illustrated in Figure 15-5 will be used throughout this chapter with the following characteristics:

- The same techniques can be applied to other deployments.
- Connection Manager LDAP will not include any users.
- Connection Manager will have access to the users in the WebSphere Everyplace Access LDAP directory.
- Connection Manager configuration will remain in its own LDAP directory.

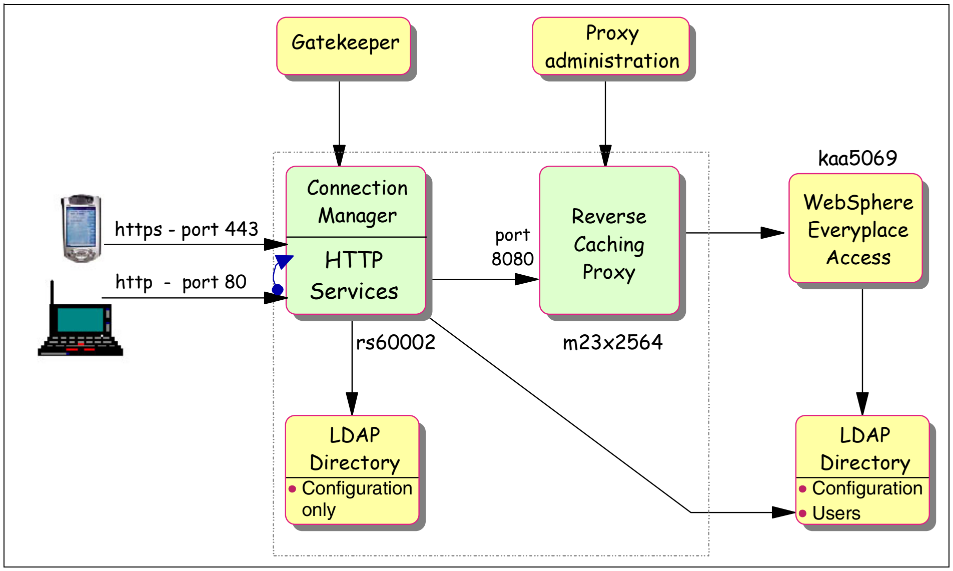


Figure 15-5 WebSphere Everyplace Access integrated sample scenario

To build this scenario five machines are used:

- Connection Manager server
  - IBM AIX version 5.0.0.2
  - IBM DB2 Universal Database
  - IBM Directory Services 4.1
  - IBM WebSphere Connection Manager 5.0
- Reverse proxy server
  - Microsoft Windows 2000 Server
  - IBM Websphere Edge Server 2.0
- WebSphere Everyplace Access server
  - Microsoft Windows 2000 Server
  - IBM Websphere Everyplace Access 4.3.0
- Gatekeeper client
  - Microsoft Windows 2000 Professional
  - IBM WebSphere Connection Manager Gatekeeper 5.0
- HTTP client
  - Microsoft Windows 2000 Professional
  - Microsoft Internet Explorer 6.0

15.2.1 Scenario setup

This scenario relies on the HTTP Access Service scenario shown in Chapter 14, “HTTP Access Services” on page 275. The scenario is illustrated using a step-by-step approach. The Gatekeeper tool is used to update the Connection Manager configuration. In addition, Connection Manager needs to be recycled after configuration changes are done, this procedure has already been described in Chapter 14, “HTTP Access Services” on page 275.

Note: For details about HTTP clients such as desktop browsers and PDAs, see Chapter 14, “HTTP Access Services” on page 275.

Create a directory services server definition

In this scenario, the Connection Manager user directory is moved from a local LDAP directory to a remote LDAP directory (in the WebSphere Everyplace Access box) so that Connection Manager and Everyplace Access will share the same users. For this reason, Connection Manager must know where the user directory will be located.

For example, follow these steps to add a directory server:

1. Log in to the Gatekeeper tool to add a directory server definition as shown in Figure 15-6. Select **Add Resource -> Directory Server**.

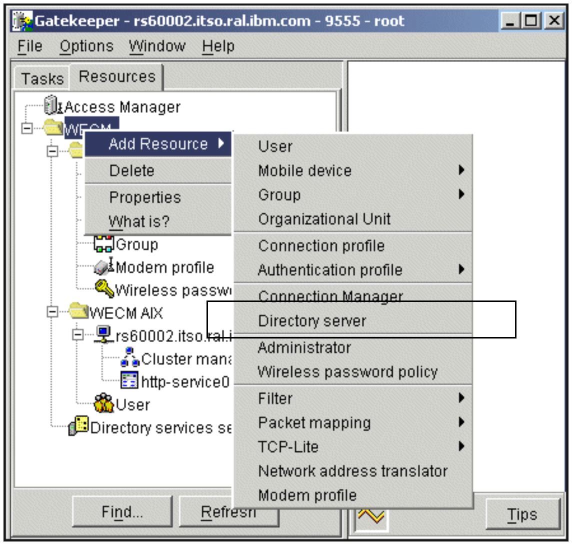
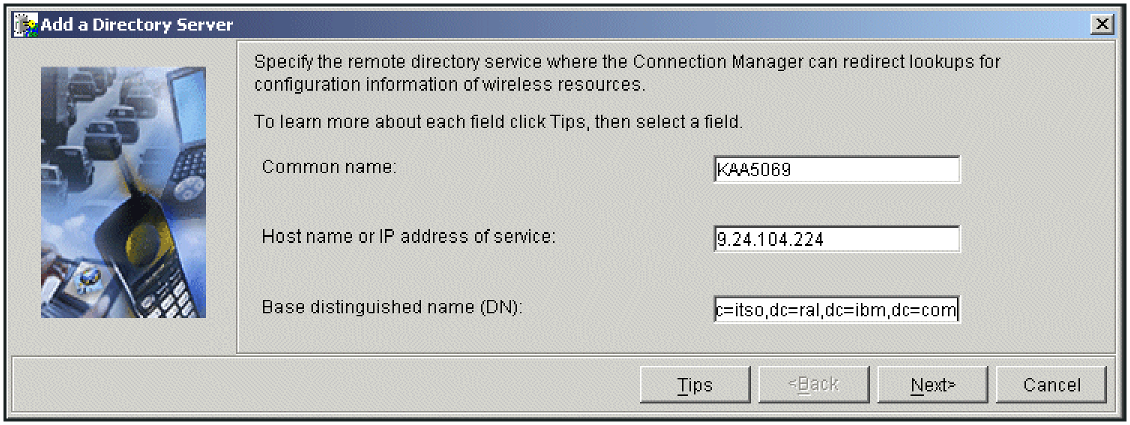


Figure 15-6 Create a directory server definition

2. Enter the common name of the resource, the hostname or IP address where LDAP is located (for this scenario LDAP is located in the WebSphere

Everyplace Access Server box), and the Base distinguished name (DN). It has to match exactly.

Note: The base distinguished name must match exactly. For example, you can use the LDAP editor DMT to double-check this information in the target LDAP directory. That is, in the Everyplace Access box for this sample scenario.



Add a Directory Server

Specify the remote directory service where the Connection Manager can redirect lookups for configuration information of wireless resources.

To learn more about each field click Tips, then select a field.

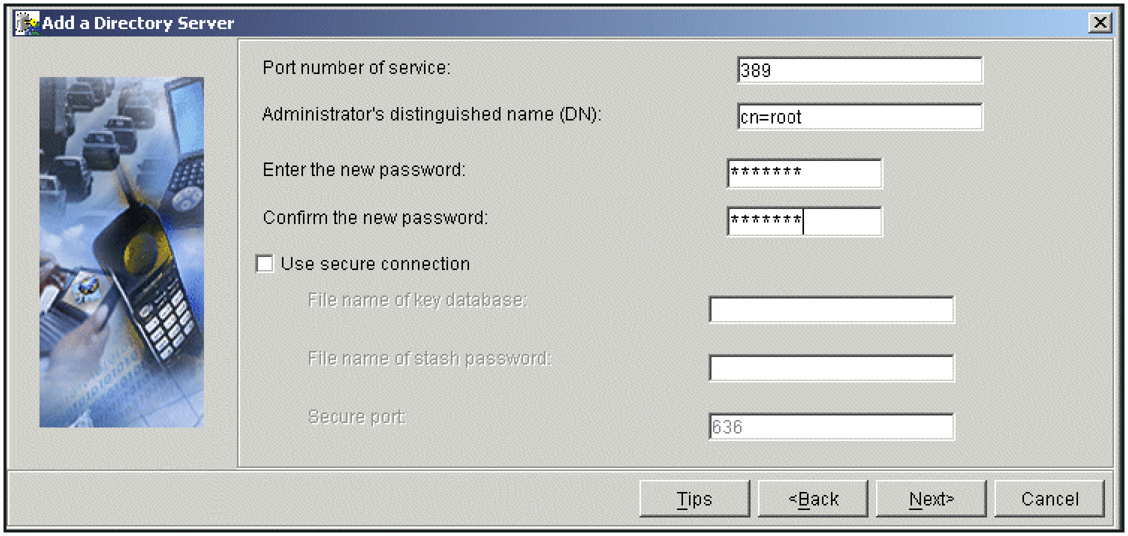
Common name: KAA5069

Host name or IP address of service: 9.24.104.224

Base distinguished name (DN): c=itso,dc=ral,dc=ibm,dc=com

Figure 15-7 Adding a Directory Server

Enter the administrator's distinguished name and password. That is, the LDAP administrator and password. This information was entered when the directory server was installed. In many cases, it is common to use cn=root as the user. Click **Next**.



Add a Directory Server

Port number of service: 389

Administrator's distinguished name (DN): cn=root

Enter the new password: *******

Confirm the new password: *******

□ Use secure connection

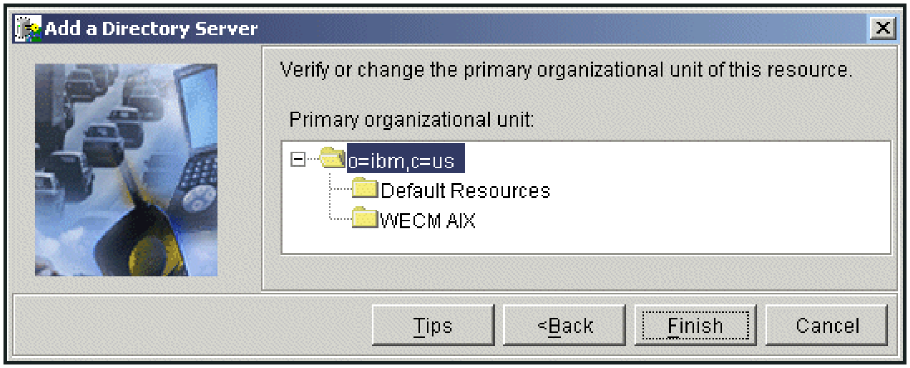
File name of key database:

File name of stash password:

Secure port: 636

Figure 15-8 Enter LDAP port number and administrator credentials

3. Verify or (if required) change the primary Organizational Unit of this resource. Click **Finish.**



Add a Directory Server

Verify or change the primary organizational unit of this resource.

Primary organizational unit:

- o=ibm,c=us
  - Default Resources
  - WECM AIX

Figure 15-9 Verify or change the primary Organizational Unit

Configure Access Manager

Once you have defined the new directory, Connection Manager needs to know which directory should be used. This option is specified in the Access Manager properties. For example:

1. Right-click **Access Manager** and select **Properties** as shown in Figure 15-10.

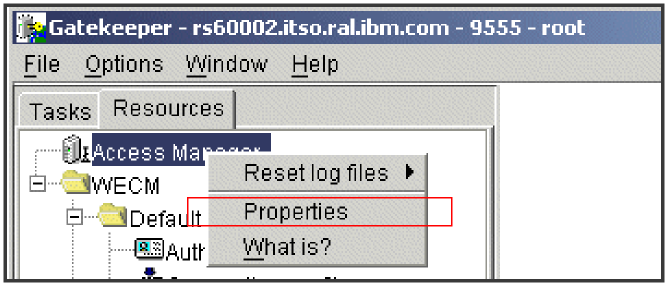
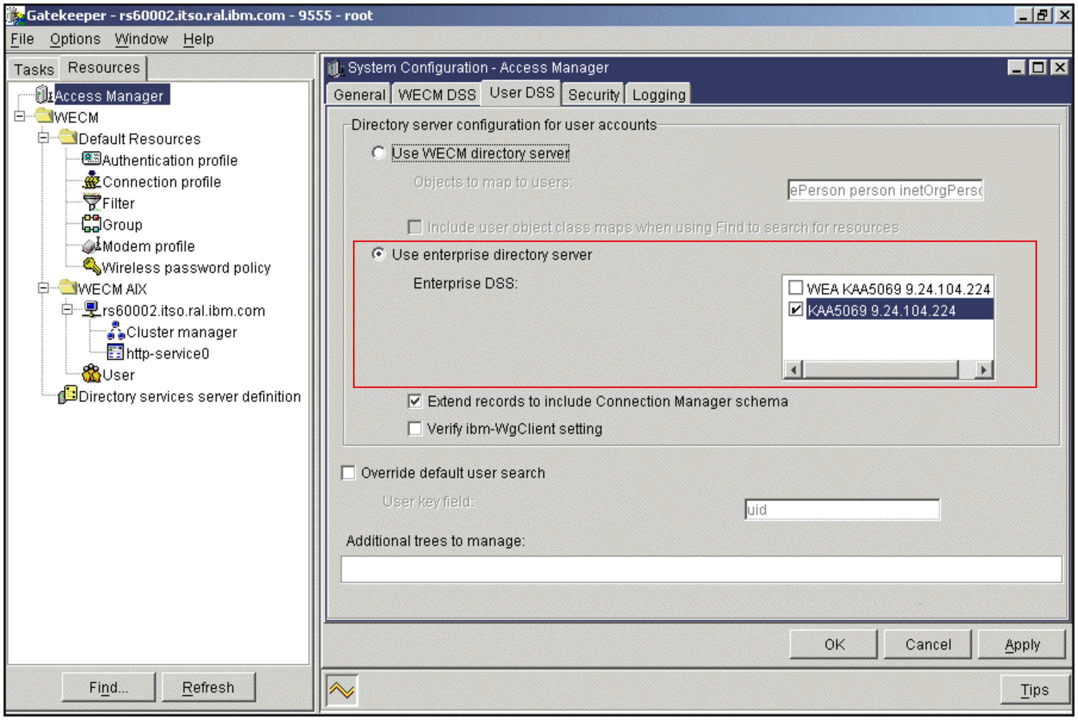


Figure 15-10 Configure Access Manger properties

2. Click the **User DSS** tab. Inside the Directory server configuration for user accounts frame select box **Use enterprise directory server** and check the DSS you created before. Click **OK.**



Directory server configuration for user accounts

○ Use WECM directory server

Objects to map to users: ePerson person inetOrgPers

□ Include user object class maps when using Find to search for resources

● Use enterprise directory server

Enterprise DSS:

□ WEA KAA5069 9.24.104.224

☑ KAA5069 9.24.104.224

☑ Extend records to include Connection Manager schema

□ Verify ibm-WgClient setting

□ Override default user search

User key field: uid

Additional trees to manage:

Figure 15-11 Enable previously created directory definition

3. At this time, you probably want to recycle Connection Manager to pick up the new configuration values.

WebSphere Everyplace Access Server configuration

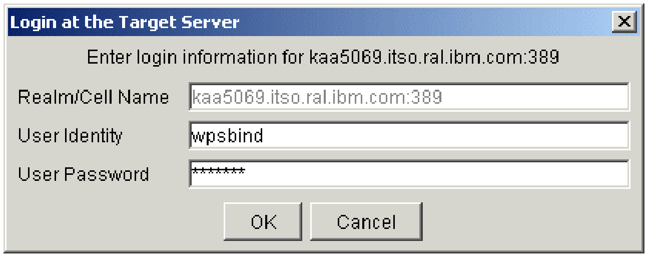
This section illustrates the following issues related to this sample scenario:

1. Check WebSphere Everyplace Access security settings.
2. Create a user to log in to Connection Manager and Everyplace Access.
3. Enable Connection Manager to access WebSphere Everyplace Access users.
4. Keep the WebSphere Everyplace Access session through Connection Manager.

Step 1: Check Security Center

You will need to check security settings, for example:

1. Select **Programs -> IBM -> WebSphere -> Application Server -> Administrator Console** and enter the user ID and password to log in.



Login at the Target Server

Enter login information for kaa5069.itso.ral.ibm.com:389

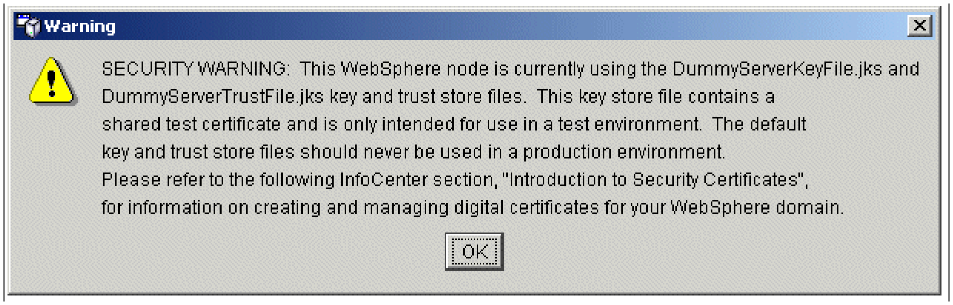
Realm/Cell Name: kaa5069.itso.ral.ibm.com:389

User Identity: wpsbind

User Password: *******

Figure 15-12 WebSphere Everyplace Access Logon

2. For testing purposes, click **OK** if you get the security alert shown in Figure 15-13.

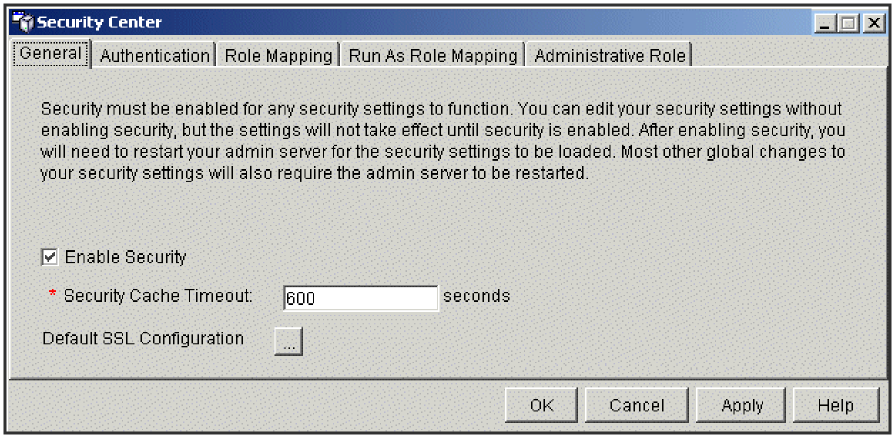


Warning

SECURITY WARNING: This WebSphere node is currently using the DummyServerKeyFile.jks and DummyServerTrustFile.jks key and trust store files. This key store file contains a shared test certificate and is only intended for use in a test environment. The default key and trust store files should never be used in a production environment. Please refer to the following InfoCenter section, "Introduction to Security Certificates", for information on creating and managing digital certificates for your WebSphere domain.

Figure 15-13 Security warning window

3. Select **Console -> Security Center**.
4. You will need to enable **Enable Security** as illustrated in Figure 15-14.



Security must be enabled for any security settings to function. You can edit your security settings without enabling security, but the settings will not take effect until security is enabled. After enabling security, you will need to restart your admin server for the security settings to be loaded. Most other global changes to your security settings will also require the admin server to be restarted.

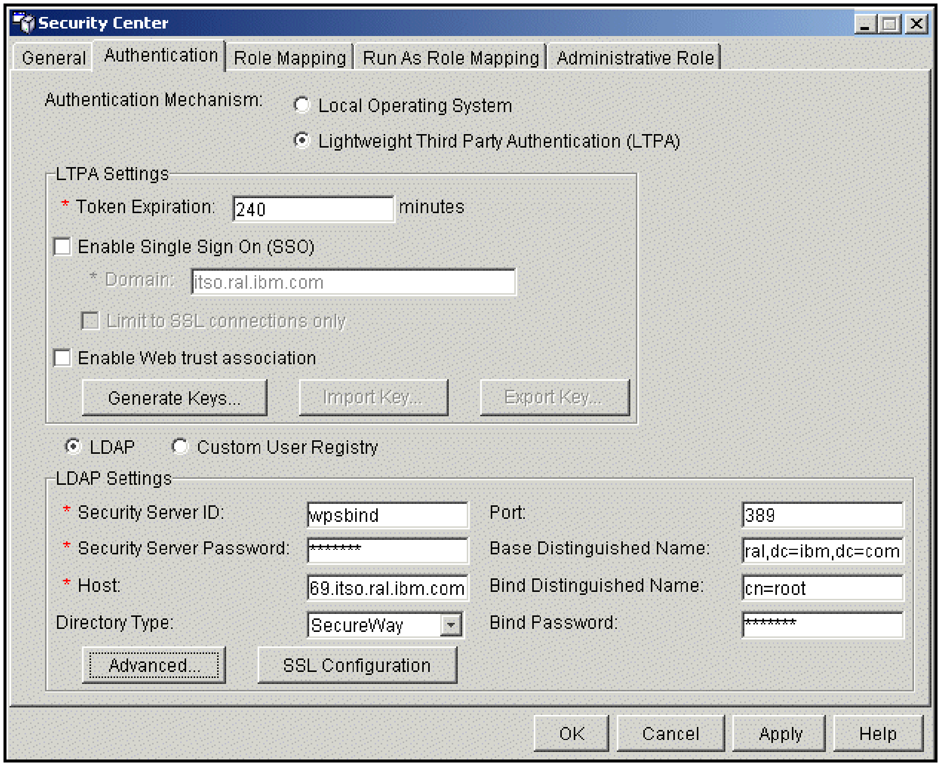
☑ Enable Security

* Security Cache Timeout: 600 seconds

Default SSL Configuration ...

Figure 15-14 Enable Security

5. Select the **Authentication** tab. See Figure 15-15.



Authentication Mechanism: ○ Local Operating System ● Lightweight Third Party Authentication (LTPA)

LTPA Settings

* Token Expiration: 240 minutes

□ Enable Single Sign On (SSO)

* Domain: itso.ral.ibm.com

□ Limit to SSL connections only

□ Enable Web trust association

● LDAP ○ Custom User Registry

LDAP Settings

* Security Server ID:	wpsbind	Port:	389
* Security Server Password:	*******	Base Distinguished Name:	ral,dc=ibm,dc=com
* Host:	69.itso.ral.ibm.com	Bind Distinguished Name:	cn=root
Directory Type:	SecureWay	Bind Password:	*******

Figure 15-15 Authentication tab

In this scenario, Single Sign-On is not required. Therefore, do not enable the check box at this time. Do not check the Enable Web trust association box either. Other parameters are default values from WebSphere Everyplace Access installation.

Step 2: Create a user

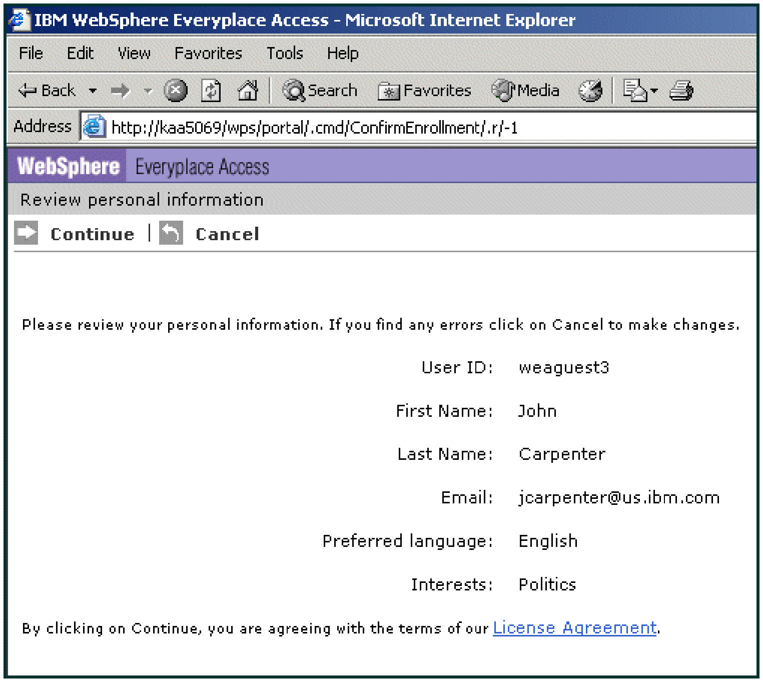
At this time you can start adding users to be accessed by both Everyplace Access and Connection Manager. For example:

1. Access the portal to create a self-enrolled user. For this sample scenario the URL is as follows:

   `http://kaa5069/wps/portal`

   where `kaa5069` is the WebSphere Everyplace Access Server name.

2. In the welcome page, click the **Sign Up** icon and enter the fields required to create a user. Figure 15-16 illustrates a sample user entered for testing.


IBM WebSphere Everyplace Access - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://kaa5069/wps/portal/.cmd/ConfirmEnrollment/.r/-1
WebSphere Everyplace Access
Review personal information
Continue | Cancel

Please review your personal information. If you find any errors click on Cancel to make changes.

User ID:	weaguest3
First Name:	John
Last Name:	Carpenter
Email:	jcarpenter@us.ibm.com
Preferred language:	English
Interests:	Politics

By clicking on Continue, you are agreeing with the terms of our License Agreement.

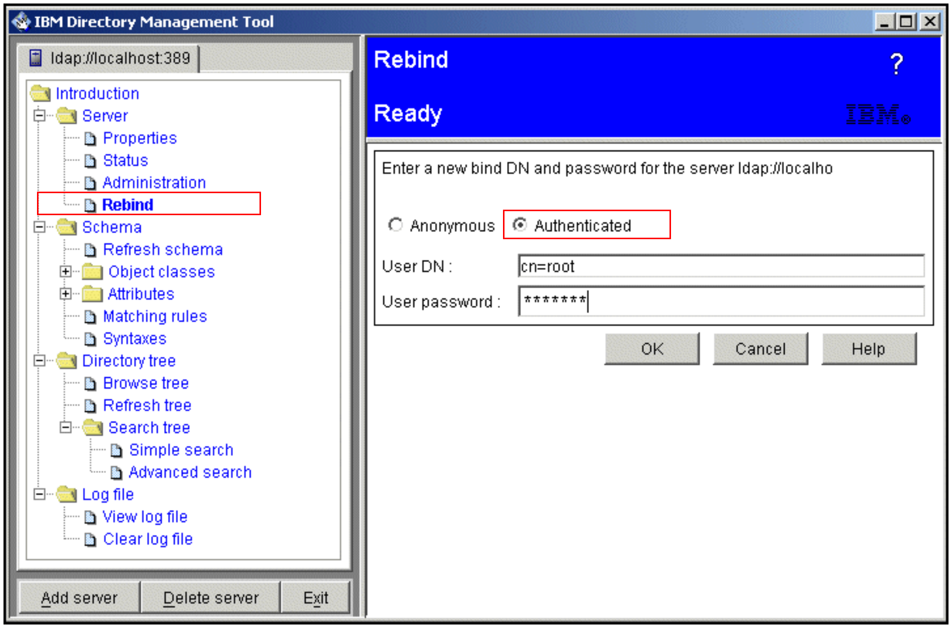
Figure 15-16 Create a user

3. Click **Continue** to confirm and finish the self-enrollment process.

Step 3: Enable Connection Manager to access users

Connection Manager should now have access to the users. However, a search of users using Gatekeeper will not show the WebSphere Everyplace Access users. The reason is because Connection Manager requires its users to have an LDAP object of class wlUser. So, it is necessary to add this object to the WebSphere Everyplace Access users. This process can be done manually or automatically. For this sample scenario the manual approach is used. For example:

1. Open DMT on the WebSphere Everyplace Access server. That is, enter `dmt` in the command line to bring up the LDAP editor.
2. In the server directory click **Rebind**. See Figure 15-17.
3. Check the **Authenticated** button and enter `User DN` and the password. Click **OK**.



Enter a new bind DN and password for the server ldap://localho

○ Anonymous ● Authenticated

User DN : cn=root

User password : *******

Figure 15-17 Server rebind

4. In the directory tree select the **dn branch** (dc=itso,dc=ral,dc=ibm,dc=com) and click **Expand** as illustrated in Figure 15-18.

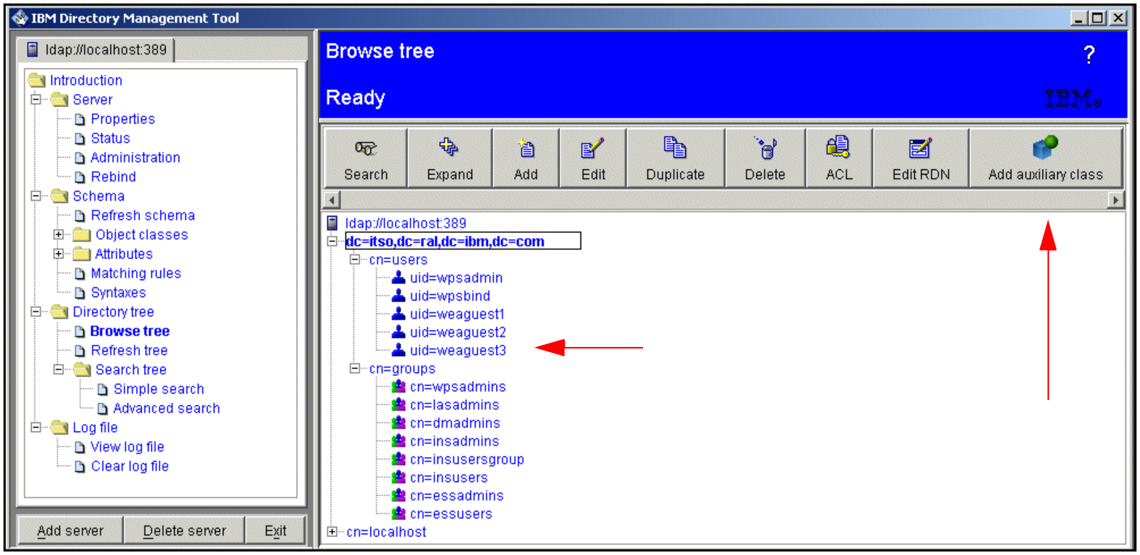
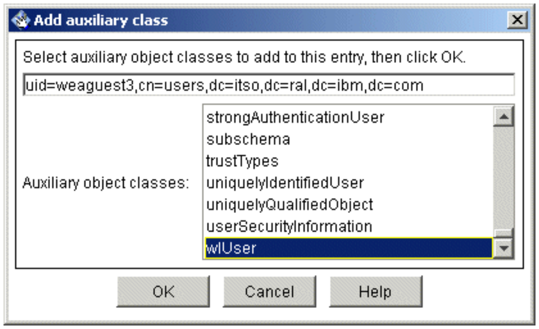


Figure 15-18 dn branch expanded

5. Select a previously created user, for example, `weaguest3`, and click the **Add auxiliary class** button as shown in Figure 15-18.
6. Select the class **wlUser** as illustrated in Figure 15-19.



Select auxiliary object classes to add to this entry, then click OK.

uid=weaguest3,cn=users,dc=itso,dc=ral,dc=ibm,dc=com

Figure 15-19 Adding wlUser class

7. Click **OK** to add the class.

As an alternative, Everyplace Access can also be configured to add the wlUser class automatically. WebSphere Portal uses a component called Manager Services used to manage users and groups. During the installation, WebSphere Portal generates configuration parameters for Member Services and stores them in {was_root}\lib\app\xml\wms.xml. You can edit this file to add the wlUser to the userObjectClass. After editing the uid section should look like this:

```
userRDNname="uid"
userMemberSubsystemAttributeName="logonId"
userObjectClass="top;inetOrgPerson;"
userDefaultBase="cn=users,dc=WebSphere Everyplace
Access,dc=rrock,dc=com"
userSearchBase="cn=users,dc=WebSphere Everyplace
Access,dc=rrock,dc=com"
```

> **Note:** Configuring WebSphere Everyplace Access to add the wlUser object automatically will result in all new users immediately enabled to be used by Connection Manager.

Step 4: Keep session going through Connection Manager

In the WebSphere Everyplace Access Welcome page, the WebSphere Everyplace Access hostname is hardcoded. So, the HTML traffic will not go through Connection Manager when you leave this page. To solve this problem (which means to have all the traffic going through Connection Manager) the hardcoded WebSphere Everyplace Access hostname should be changed to the Connection Manager hostname.

The next steps illustrates how to do this. For example, to change the hostname:

1. Open the file ConfigService.properties. Its location in a Windows machine is:

   ```
   C:\WebSphere\AppServer\lib\app\config\services
   ```

   Its location in a Linux/AIX machine is:

   ```
   /usr/webSphere/AppServer/lib/app/config/services
   ```

2. Change the WebSphere Everyplace Access hostname with the Connection Manager host name rs60002.itso.ral.ibm.com in this sample scenario as shown in Example 15-1.

Example 15-1 Hostname

```
# The parameters of the (virtual) host that the portal is accessed through
#
# Default: localhost (host.name)

host.name       =rs60002.itso.ral.ibm.com
host.port.http  =80
host.port.https =
```

Delete old JavaServer Pages

3. You will need to delete the old JavaServer Pages. This will allow Everyplace Access to create new pages with the new host name. For example, as shown in Figure 15-20, delete all files inside the wps.war directory. Its location in a Windows machine is:

   ```
   C:\WebSphere\AppServer\temp\kaa5069\WebSphere_Portal\WPS_Enterprise_Applica
   tion\wps.war
   ```

   Its location in a Linux/AIX machine is:

   ```
   /usr/WebSphere/AppServer/temp/kaa5069/WebSphere_Portal/WPS_Enterprise_Appli
   cation/wps.war
   ```

   where `kaa5069` is the WebSphere Everyplace Access server name.

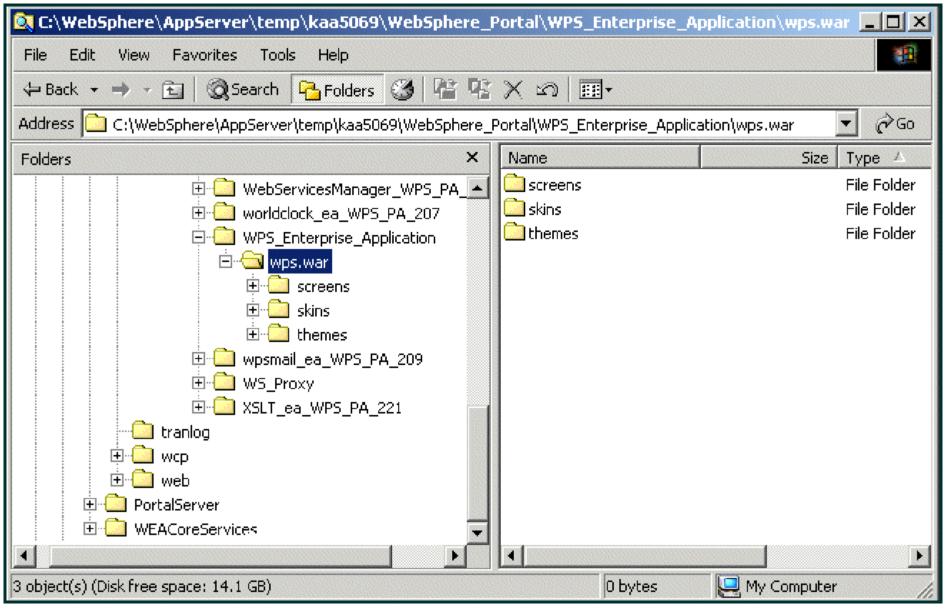

C:\WebSphere\AppServer\temp\kaa5069\WebSphere_Portal\WPS_Enterprise_Application\wps.war
File Edit View Favorites Tools Help
Back Search Folders
Address C:\WebSphere\AppServer\temp\kaa5069\WebSphere_Portal\WPS_Enterprise_Application\wps.war
Go
Folders
WebServicesManager_WPS_PA_
worldclock_ea_WPS_PA_207
WPS_Enterprise_Application
wps.war
screens
skins
themes
wpsmail_ea_WPS_PA_209
WS_Proxy
XSLT_ea_WPS_PA_221
tranlog
wcp
web
PortalServer
WEACoreServices
Name Size Type
screens File Folder
skins File Folder
themes File Folder
3 object(s) (Disk free space: 14.1 GB)
0 bytes
My Computer

Figure 15-20 Delete old JavaServer Pages

4. Recycle WebSphere Everyplace Access to pick up the updates.

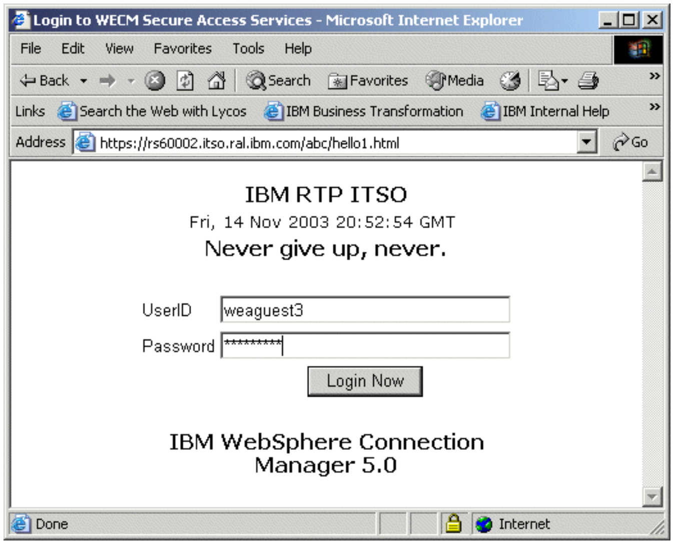
15.2.2 Running the sample scenario

When you run the sample scenario, log in to Connection Manager using the Everyplace Access user, which has also been enabled for Connection Manager. For example, using Connection Manager HTTP Services:

1. Access Everyplace Access portal by connecting to Connection Manager using the following URL:

```
https://rs60002.itso.ral.ibm.com/wps/myportal
```

2. When receiving a certificate signed by an unrecognized Certificate Authority (CA), for example, a self-signed certificate, you will receive a security alert. For testing purposes, accept the security alert.
3. As shown in Figure 15-21, log in to Connection Manager with a valid user. Click **Login Now**.


Login to WECM Secure Access Services - Microsoft Internet Explorer

Address: https://rs60002.itso.ral.ibm.com/abc/hello1.html

IBM RTP ITSO

Fri, 14 Nov 2003 20:52:54 GMT

Never give up, never.

UserID: weaguest3

Password: *********

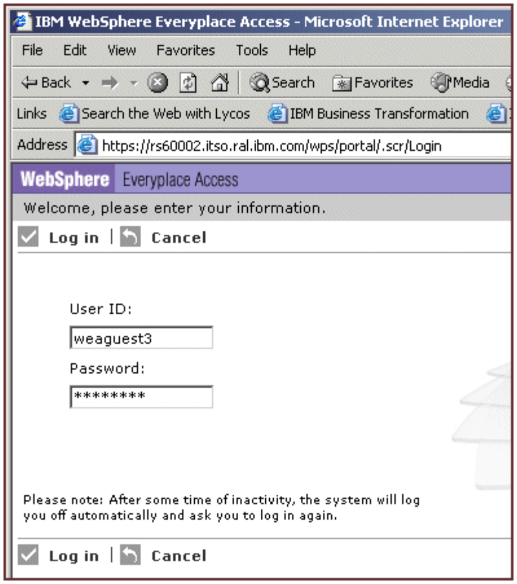
Login Now

IBM WebSphere Connection Manager 5.0

File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Links
Search the Web with Lycos
IBM Business Transformation
IBM Internal Help
Go
Done
Internet

Figure 15-21 Connection Manager integrated login

4. Once you are authenticated by Connection Manager, you will have access to the Everyplace Access welcome page. Log in to WebSphere Everyplace Access with the same user and password as shown in Figure 15-22.


IBM WebSphere Everyplace Access - Microsoft Internet Explorer

Address https://rs60002.itso.ral.ibm.com/wps/portal/.scr/Login

Welcome, please enter your information.

User ID: weaguest3

Password: ********

Please note: After some time of inactivity, the system will log you off automatically and ask you to log in again.

Figure 15-22 WebSphere Everyplace Access login

Although in this scenario, WebSphere Everyplace Access is sharing users with Connection Manager, you will still need to log in initially to Connection Manager and then to Everyplace Access. This situation is solved by implementing Single Sign-On using the Lightweight Third Party Authentication (LTPA). For details about Single Sign-On, see Chapter 16, “Single Sign-On” on page 323, and for a sample scenario see Chapter 17, “WebSphere Everyplace Access Single Sign-On using LTPA” on page 347.

16

Single Sign-On

This chapter provides information relating to establishing Single Sign-On (SSO) between Connection Manager and other suitable applications. SSO provides a way of enhancing usability by reducing the number of authentication challenges that are presented to a user. In a Connection Manager environment, it is possible to configure it such that the only time a user has to enter their credentials is at the initial challenge provided by Connection Manager.

The topics discussed in this chapter include:

- Trust Association Inteceptor (TAI)
- Lightweight Third Party Authentication (LTPA)
- Configuring SSO between Connection Manager and WebSphere Application Server
- Configuring SSO between Connection Manager and Domino

16.1 Overview

Many Connection Manager environments include at least one application server used to process and serve data to a range of clients. These applications may require the user to be authenticated prior to sending any requests. WebSphere Everyplace Access is an example of such an application whereby the user is presented with a challenge prior to being authenticated in the system.

In a standard Connection Manager environment, such a challenge by an application means that the user has to re-enter their credentials despite already having done so in order to log in to the Connection Manager. For the sake of enhanced usability, it is recommended to try and avoid this type of double authentication.

This section will discuss the techniques that can be applied in a Connection Manager environment to establish a relationship between Connection Manager and downstream applications servers, thus creating single sign-on. With SSO, the user is only required to supply credentials for the initial Connection Manager login. Any following request to appropriately configured application servers will recognize the user has already been authenticated, and treat the requests accordingly.

16.2 SSO and WebSphere Application Server

Between Connection Manager V5 and WebSphere Application Server, two different methods are supported for establishing single sign-on. These are:

- Trust Association Inteceptor (TAI)
- Lightweight Third Party Authentication (LTPA)

In order to provide the user with single sign-on each access method requires a way for the user to acquire an LTPA token after being authenticated by Connection Manager.

16.2.1 Trust Association Inteceptor (TAI)

Connection Manager V5 provides a TAI plug-in for use with WebSphere Application Server. This allows SSO from Mobility Clients connecting to WebSphere Application Server through a VPN. Clients using this feature for SSO must be registered both as a Connection Manager user, and as a user of the WebSphere Application Server applications.

The TAI plug-in is installed into WebSphere Application Server. It intercepts HTTP requests and extracts the source IP address, which relates to the VPN IP assigned to the client by Connection Manager. Using this IP address, a search is made of Connection Manager's active session table looking for the user associated to that address. If found, the TAI builds a fully qualified LDAP user DN to represent how the user is know to WebSphere Application Server. This is based on a predefined template. The TAI then forwards the user DN to WebSphere Application Server to be used for verifying access to WebSphere Application Server applications. Once the credentials have been verified, an LTPA token is generated by the WebSphere Application Server security subsystem. This token will be stored by the client and sent with each subsequent request as part of the HTTP header.

> **Note:** The TAI plug-in expects the source IP address to match that assigned to the Mobility Client. Any component that may alter the source IP as seen by the TAI plug in, such as a proxy or NAT router, will cause the lookup to fail.

If separate directory servers are used for Connection Manager and WebSphere Application Server, this process requires that the user ID (as known to Connection Manager for a given user) must match that stored by WebSphere Application Server. As the TAI relies on the VPN address assigned by Connection Manager as its key, care should also be taken to ensure that these addresses cannot be spoofed by non-VPN users.

16.2.2 Lightweight Third Party Authentication (LTPA)

Connection Manager is able to generate LTPA tokens that may be used as the mechanism for establishing SSO between Connection Manager and other supported applications for HTTP Access Services using HTTP access client authentication method. This is supported by both RADIUS and LDAP-bind type authentication profiles.

When enabled, an LTPA token is generated when a HTTP based user is authenticated by a Connection Manager. This token is stored in a browser cookie to support SSO with other LTPA enabled application servers.

All servers using Connection Manager's LTPA support must have the same LTPA keys and password, and reside in the same DNS domain. The key may be generated either through the Gatekeeper, or through a suitable application servers console such as WebSphere Application Server. This process is discussed further in Chapter 17, "WebSphere Everyplace Access Single Sign-On using LTPA" on page 347.

While LTPA tokens are also created when using the TAI plugin for SSO, the difference here is that the token is generated by Connection Manager rather than a downstream application.

16.3 Using TAI

The TAI is especially useful for enabling SSO when connected to Connection Manager through a Mobility Client. The following section describes how to configure the TAI, and verify that it is working correctly.

For the purpose of this discussion, we configure SSO between Connection Manager and WebSphere Everyplace Access. The WebSphere Everyplace Access applications are running on WebSphere Application Server V4.0.4 on a Windows 2000 host.

16.3.1 Configuring the TAI plug-in

The TAI is currently not supplied with Connection Manager V5 as it is available as a separate download. For example, inside the TAI plugin ZIP file you will find the following components:

- wecm-tai.jar

  The jar file containing the plug-in classes
- trustedservers.properties.sample

  Sample trusted server property file, which is supplied to WebSphere Application Server
- wecm.properties.sample

  Sample Connection Manager properties file

Refer to the following steps for installation:

Initial steps

1. Ensure LDAP, DB2, and Connection Manager V5 are installed and configured correctly as per the installation chapter.
2. The TAI plugin will read information from the Connection Manager session database. If this database is remote to the WebSphere Application Server, a local alias will need to be created on the WebSphere Application Server.
   a. Start the DB2 command interface:
      - UNIX: Execute a **su** task to the WebSphere Application Server database instance owner, then run the command **db2**
      - Windows: As an administratoor user, run **Start -> IBM DB2 -> Commmand Line Processor.**

   b. At the prompt, run the following command to catalog a remote node:

```
catalog tcpip node <node_name> remote <WECM_host> server <WECM_DB_port>
```

where:

- `<node_name>` is the name you chose for the node.
- `<WECM_host>` is the hostname for the Connection Manager server.
- `<WECM_DB_port>` is the port on which the Connection Manager session database listens.

The node can be verified by running **`list node directory`** from the command prompt. The output will be similar to Example 16-1.

Example 16-1 Node list

```
Node 4 entry:

 Node name                      = WECM
 Comment                        =
 Protocol                       = TCPIP
 Hostname                       = sun4
 Service name                   = 50000
```

c. At the prompt, run the following command to catalog the remote database at the node created above:

```
catalog db <WECM_DB> as <WECM_DB> at node <node_name>
```

where:

- `<WECM_DB>` is the name of the Connection Manager session database. By default, this will be **`wgdata`**.

The database details can be viewed by running **`list db directory`**. The output will look similar to Example 16-2.

Example 16-2 Database directory

```
Database 12 entry:

 Database alias                  = WGDATA
 Database name                   = WGDATA
 Node name                       = WECM
 Database release level          = 9.00
 Comment                         =
 Directory entry type            = Remote
 Catalog node number             = -1
```

d. Test the connection by running the following command:

```
connect to <WECM_DB> user <WECM_DB_USER> using <WECM_DB_PW>
```

where:

- `<WECM_DB_USER>` is the user ID used by Connection Manager to connect to the database
- `<WECM_DB_PW>` is the password used to connect to the Connection Manager database

If successful, the output should look similar to that in Example 16-3.

Example 16-3   Successful database connection

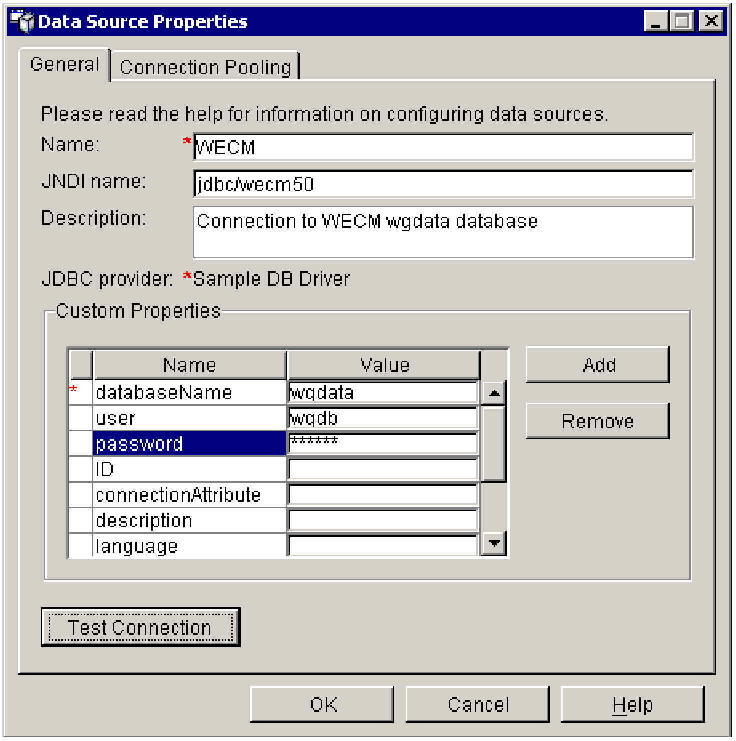
```
db2 => connect to wgdata user wgdb using ibmdb2

   Database Connection Information

 Database server        = DB2/SUN 7.2.8
 SQL authorization ID   = WGDB
 Local database alias   = WGDATA

db2 =>
```

Once all has been verified, exit the DB2 interface by entering **`quit`** at the prompt.

Configuring for WebSphere Application Server V4

The following configuration steps need to be completed in WebSphere Application Server to install the Connection Manager TAI plug in:

1. Start the WebSphere Application Server administration console:
   – UNIX: From the directory <WAS_HOME/bin, run the command **`./adminclient.sh <node_name> <admin_port>`**, where <WAS_HOME> is the WebSphere Application Server installation root directory.
   – Windows: Click **Start -> Programs -> IBM WebSphere -> Application Server V4.0 -> Administrator's Console**
2. Create a data source to allow connections from WebSphere Application Server to the database configured in “Initial steps” on page 326. Either create a new DB2 JDBC provider for this data source, or use the sample JDBC provider as described below:
   a. From the admin console, select **Resources -> JDBC Providers -> Sample DB Driver -> Datasources.**
   b. Right-click **Datasources** and select **New...**
   c. Add the details similar to that shown in Figure 16-1. The database name is the one used in “Initial steps” on page 326. Choose any suitable name and JNDI name for the data source.



Please read the help for information on configuring data sources.

Name: *WECM

JNDI name: jdbc/wecm50

Description: Connection to WECM wgdata database

JDBC provider: *Sample DB Driver

Custom Properties

	Name	Value
*	databaseName	wgdata
	user	wgdb
	password	******
	ID	
	connectionAttribute	
	description	
	language	

Figure 16-1 New data source properties

d. Click **Test Connection** to verify the status of the new connection. If the test is not successful, review and correct the settings provided. Once the test is verified as working, click **OK** to continue.

3. Enable global security using LTPA:

a. Click **Console -> Security Center...**

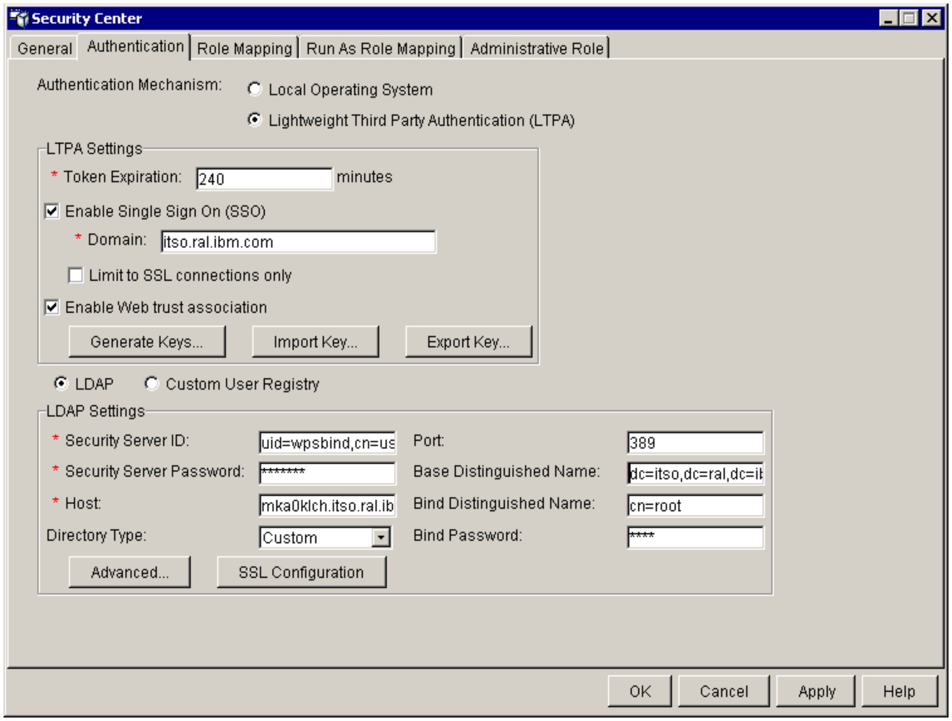
b. On the **General** tab, ensure **Enable Security** is checked.

c. Select the **Authentication** tab and complete the following steps:

- Select **Lightweight Third Party Authentication (LTPA)** as the authentication mechanism
- Set an appropriate value for the LTPA token expiration.

> **Note:** When the token is set in the Web browser, it will expire when the browser is closed. However, if the browser is running longer than the token expiry time, a new token will be generated on expiry.

- Check **Enable Single Sign On (SSO)** and enter the domain name. All servers requiring SSO must be in the same domain.
- Check **Enable Web trust association.**
- Check **LDAP** as the user registry type.
- Complete the LDAP settings as per the directory server configuration. The Connection Manager LDAP may be used for the user repository, or some other LDAP may be selected. As per Figure 16-2, the LDAP used was that used by the WebSphere Everyplace Access server.



Authentication Mechanism: ○ Local Operating System ● Lightweight Third Party Authentication (LTPA)

LTPA Settings

* Token Expiration: 240 minutes

☑ Enable Single Sign On (SSO)

* Domain: itso.ral.ibm.com

□ Limit to SSL connections only

☑ Enable Web trust association

● LDAP ○ Custom User Registry

LDAP Settings

* Security Server ID:	uid=wpsbind,cn=us	Port:	389
* Security Server Password:	*******	Base Distinguished Name:	dc=itso,dc=ral,dc=ib
* Host:	mka0klch.itso.ral.ib	Bind Distinguished Name:	cn=root
Directory Type:	Custom	Bind Password:	****

Figure 16-2 Authentication settings

d. Click **Apply.**

e. Stop WebSphere Application Server.

f. Unzip the wecm-tai.zip file to a temporary directory.

g. From the temporary directory used in step f, copy wecm-tai.jar to <WAS_HOME>/lib.

h. Edit <WAS_HOME>/properties/trustedserver.properties to match the sample supplied with the Connection Manager TAI. Ensure the following lines are correct:

- `com.ibm.websphere.security.trustassociation.types=wecm`
- `com.ibm.websphere.security.trustassociation.wecm.interceptor=com.ibm.wecm.security.tai.WecmTrustAssociationInterceptor`
- `com.ibm.websphere.security.trustassociation.wecm.config=wecm`

i. Copy wecm.properties.sample from the temporary directory to <WAS_HOME>/properties/wecm.properties.

j. Edit <WAS_HOME>/properties/wecm.properties, adding the appropriate properties for the environment. If required, use Example 16-4 as a guide. The user template field is optional, and not required if the fully qualified user DN of the Connection Manager LDAP matches that used by WebSphere Application Server for testing a user ID. On the first successful load of the Connection Manager TAI, WebSphere Application Server will modify this file by encrypting each of the displayed passwords.

Example 16-4 wecm.properties configuration

```
# WECM Trust Association Interceptor Configuration Properties

# WECM Version
com.ibm.wecm.security.tai.wecmversion=500

# User template
com.ibm.wecm.security.tai.usertemplate=uid={0},cn=users,dc=itso,dc=ral,dc=ibm,d
c=com

# WECM Directory Server Configuration

# LDAP server URL
com.ibm.wecm.security.tai.wecmldapurl=ldap://rs600035.itso.ral.ibm.com:389
# LDAP bind user id
com.ibm.wecm.security.tai.wecmldapuser=cn=root
# LDAP bind password
com.ibm.wecm.security.tai.wecmldappassword=secret
# WECM LDAP base suffix
com.ibm.wecm.security.tai.wecmsuffix=dc=itso,dc=com

# Version specific configuration properties below

# Version 5 - active session table DB configuration
# WECM datasource JNDI location
com.ibm.wecm.security.tai.datasource=jdbc/wecm
# WECM database user id
com.ibm.wecm.security.tai.dbuser=wgdb
```

```
# WECM database password
com.ibm.wecm.security.tai.dbpassword=secret

# Version 4 - LDAP user registry
# Suffix to search for WECM user records (if different from base suffix)
# com.ibm.wecm.security.tai.usersuffix=cn=users,ou=IBM,dc=raleigh,dc=ibm,dc=com
```

k. Start WebSphere Application Server. The TAI plugin is now installed. To verify this, view the wecm.properties files. If successful, the passwords will have been encrypted, and the layout will look slightly different.

Configuring for WebSphere Application Server V5.0.1

To configure the Connection Manager TAI for use in WebSphere Application Server V5, complete the following steps:

1. Enable and configure WebSphere Application Server global security to use LTPA:

   a. From the WebSphere Application Server administration console, select **Security -> User Registries -> LDAP**. Enter valid details relating to the LDAP server you will be using. Click **OK**.

   b. Select **Security -> Authentication Mechanisms -> LTPA**. Either import existing LTPA keys, or generate new ones. Click **OK** when done.

   c. Select **Security -> Global Security** and verify the following items are set:

      - Check the box to enable global security.
      - Select **LTPA** as the **Active Authentication Mechanism.**
      - Select **LDAP** and the **Active User Mechanism**.

      Click **OK** when done.

   d. Create a DB2 JDBC provider:

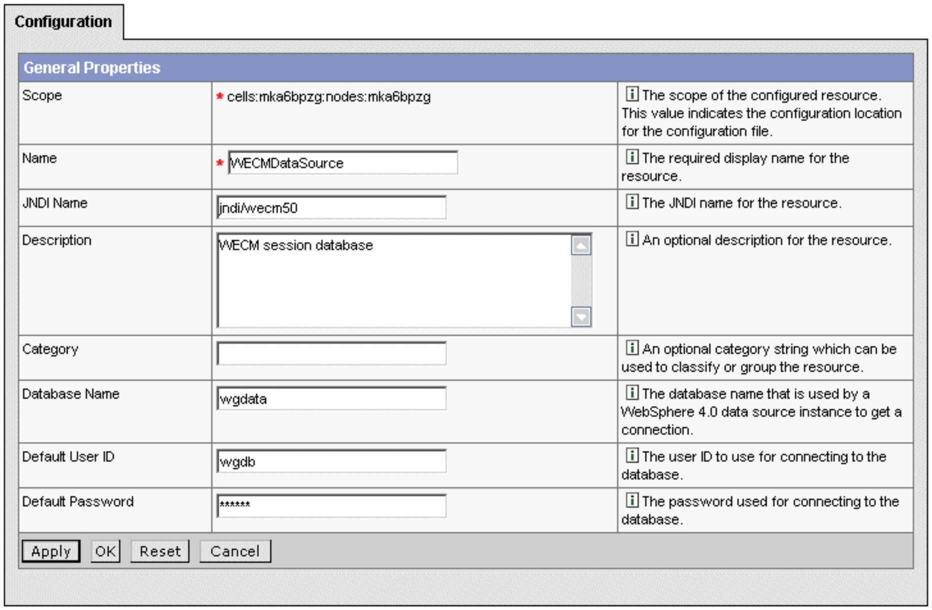
      i. Select **Resources -> JDBC Providers**.

      ii. Click **New**.

      iii. For DB2 V7.2 databases, chose **DB2 Legacy CLI-based Type 2 JDBC Driver**. For other type of databases, consult the WebSphere Application Server InfoCenter at:
      `http://www-3.ibm.com/software/webservers/appserv/infocenter.html`

      iv. Click **OK**. After verifying the configuration settings, and then click **Data Sources (version 4)** to add a data source to the new provider.

      v. Click **New**, and enter the relevant data source details. The database name is the one used in “Initial steps” on page 326. Choose any suitable name and JNDI name for the data source. Figure 16-3 on page 333 shows an example of the configuration. Click **OK** when complete.



Configuration

General Properties

Scope	* cells:mka6bpzg:nodes:mka6bpzg	The scope of the configured resource. This value indicates the configuration location for the configuration file.
Name	* WECMDataSource	The required display name for the resource.
JNDI Name	jndi/wecm50	The JNDI name for the resource.
Description	WECM session database	An optional description for the resource.
Category		An optional category string which can be used to classify or group the resource.
Database Name	wgdata	The database name that is used by a WebSphere 4.0 data source instance to get a connection.
Default User ID	wgdb	The user ID to use for connecting to the database.
Default Password	******	The password used for connecting to the database.

Figure 16-3 Data source settings

vi. Click **Save** to save your changes.

e. Stop WebSphere Application Server.

f. Copy wecm.properties.sample from the temporary directory to <WAS_HOME>/properties/wecm.properties.

g. Edit <WAS_HOME>/properties/wecm.properties, adding the appropriate properties for the environment. Ensure the JNDI name matches that entered when creating the data source. If required, use Example 16-4 on page 331 as a guide. The user template field is optional, and not required if the fully qualified user DN of the Connection Manager LDAP matches that used by WebSphere Application Server for testing a user ID. On the first successful load of the Connection Manager TAI, WebSphere Application Server will modify this file by encrypting each of the displayed passwords.

h. Start WebSphere Application Server.

i. Select **Resources -> JDBC Providers** and click your JDBC provider used for Connection Manager.

j. Tick the box next to the **Connection Manager data source** and click **Test Connection**. This will ensure the data source is functioning correctly before proceeding.

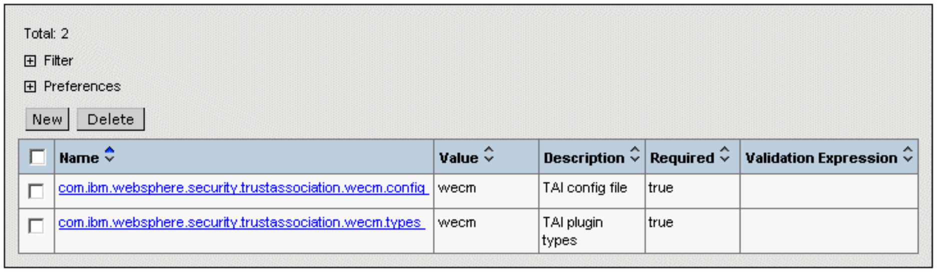
k. Copy wecm-tai.jar to <WAS_HOME>/lib.

l. Select **Security -> Authentication Mechanisms -> LTPA -> Trust Association**. Enable trust association, and then click **Interceptors**.

m. Click **New** to add a new interceptor class.

n. Enter the class name `com.ibm.wecm.security.tai.WecmTrustAssociationInterceptor`, then click **Apply**.

o. Click **Custom properties**, and create a property named `com.ibm.wecm.security.trustassociation.types` and set its value to `wecm`. Create a second property named `com.ibm.websphere.security.trustassociation.wecm.config`, also with a value set to `wecm`. The result should be the same as in Figure 16-4.



□	Name	Value	Description	Required	Validation Expression
□	com.ibm.websphere.security.trustassociation.wecm.config	wecm	TAI config file	true	
□	com.ibm.websphere.security.trustassociation.wecm.types	wecm	TAI plugin types	true	

Figure 16-4 Custom properties

p. Save your changes and restart WebSphere Application Server. The TAI should now be installed.

16.3.2 Verifying SSO through the TAI plugin

The following describes the method used to verify SSO using the TAI plugin. The configuration used for this example included the following items, and is displayed in Figure 16-5:

- Connection Manager V5 Win32 Mobility Client
  - Configured to use password key exchange (default), and is able to access the required Connection Manager.
- Connection Manager V5

  - The Connection Manager used for this test uses the default installed authentication method.
- WebSphere Application Server V4.0.4
  - Required by WebSphere Everyplace Access
  - TAI plugin installed and configured as per “Configuring the TAI plug-in” on page 326
- WebSphere Everyplace Access V4.3
  - Used to demostrate SSO. This was chosen as it requires user authentication.
- IBM Directory Services V4.1
  - Separate instances configured for both Connection Manager and WebSphere Everyplace Access.
- DB2 UDB V7.2
  - Required by Connection Manager, WebSphere Application Server, and IBM Directory Services

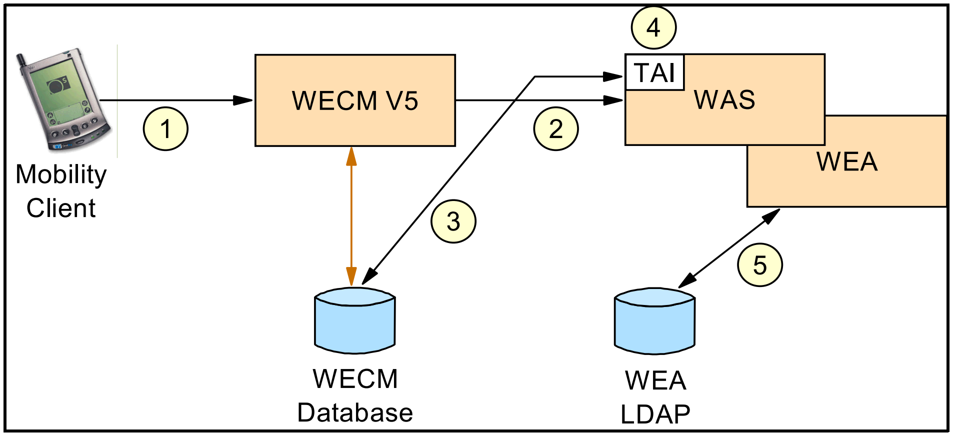


Figure 16-5 Example configuration using Connection Manager TAI plugin

The items below describe the transaction flow used in Figure 16-5:

1. A Mobility Client based user logs into the Connection Manager. At that time Connection Manager creates an entry in its session table, which include the clients user ID and the VPN IP address assigned to that client.
2. Through the clients browser, a request is made to WebSphere Everyplace Access.
3. This request is intercepted by the TAI plugin. The plugin extracts the source IP of the request, which in this case is the client’s VPN IP address. The plugin

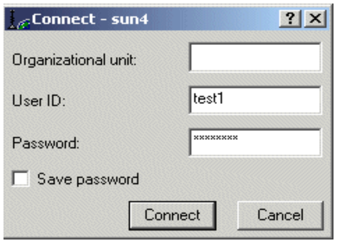
then searches the Connection Manager session table to see whether or not this IP address is assigned to an active session. If so, the client's user ID is returned.

4. The TAI uses this user ID and builds a fully qualified user DN based on the template defined in wecm.properties as discussed in Example 16-4 on page 331. This is then passed to WebSphere Application Server for use by relevant application servers.
5. WebSphere Everyplace Access applications use supplied user DN to authenticate the user. If the user exists in its user repository, the user will be allowed to continue without being required to again enter their credentials.

The following steps were performed to verify SSO:

1. Start the Mobility Client, and enter the user details.

> **Note:** For this scenario, the user must exist in both the Connection Manager user directory, and the WebSphere Everyplace Access user directory.


Connect - sun4

Organizational unit:

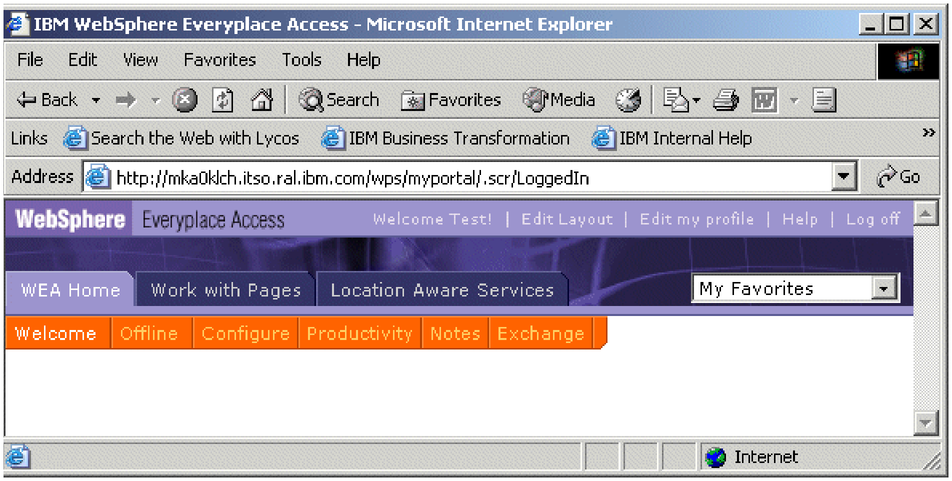
User ID: test1

Password: *********

□ Save password

Figure 16-6 Mobility Client logon

2. Click **Connect**.
3. Once connected, use a Web browser to navigate to `http://<WEA_Host>/wps/myportal`.
4. Typically, without SSO enabled, the user would again be challenged by WebSphere Everyplace Access. With SSO, the second challenge is handled by the TAI plugin and, if successful, the request is forwarded to the target with no further user interaction. A page similar to Figure 16-7 is displayed.



Address: http://mka0klch.itso.ral.ibm.com/wps/myportal/.scr/LoggedIn

Figure 16-7 Single sign-on to WebSphere Everyplace Access

16.4 Using LTPA with HTTP Access Services

The following section describes how to configure and use LTPA to enable SSO between Connection Manager and LTPA aware services such as WebSphere Application Server.

Note: For details about Single Sign-On with WebSphere Everyplace Server, see also Chapter 17, “WebSphere Everyplace Access Single Sign-On using LTPA” on page 347.

16.4.1 Configuring LTPA

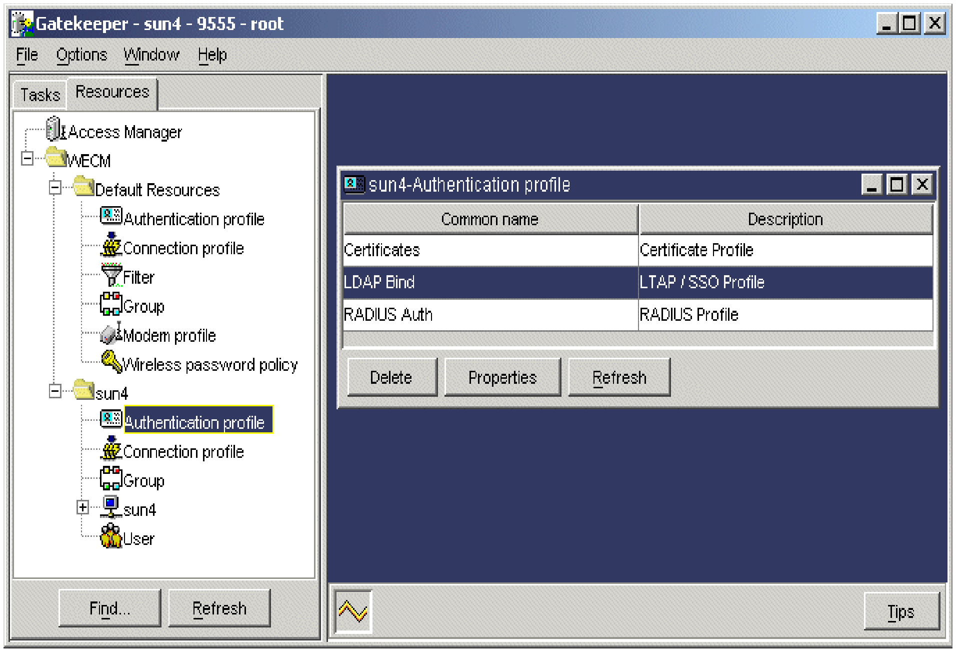
To enable Connection Manager to generate LTPA tokens for SSO, a suitable authentication profile must first be defined. The following describes how to enable LTPA/SSO in the relevant authentication profile. The method is the same for both LDAP-bind and RADIUS type authentication. When using LTPA between the Connection Manager and a WebSphere Application Server or portal server environment, be aware of the following:

- The Access Manager should be configured to not encrypt passwords prior to storing.
- WebSphere Application Server and portal server are able to use a user ID defined by Connection Manager, however by default, Connection Manager cannot use user IDs defined by WebSphere Application Server and portal server. To allow this, the schema of the directory used by WebSphere

Application Server and the portal server must be extended to include the Connection Manager object classes. This may be done by importing the relevant LDIF file found in <WECM_HOME>/conf.

Prior to completing the following steps, ensure that at least one of these types of authentication profiles exist:

1. From the Gatekeeper, select either the LDAP-bind or RADIUS authentication profile created earlier similar to as shown in Figure 16-8. The profile used for this discussion is LDAP-bind, however, the process is identical for RADIUS. Click **Properties**.



Common name	Description
Certificates	Certificate Profile
LDAP Bind	LTAP / SSO Profile
RADIUS Auth	RADIUS Profile

Figure 16-8 Authentication profile selection

2. On the properties page, select the **LTPA/SSO** tab.
3. Tick the box to indicate **LTPA Enabled**. Either accept the default value of 240 minutes, or enter an alternate value for **LTPA token lifetime**. This indicates the amount of time the token will be valid for from the time it was created, regardless of the length of the user's session.
4. Tick the box to indicate **SSO Enabled**. Enter the SSO domain in the box provided. This relates to the DNS domain in which SSO will operate. All parties involved in SSO must exist in the same DNS domain.

5. For initial configuration, the LTPA keys must be either generated and the exported to another LTPA aware application, or imported from a file. To generate the keys:

    a. Select **Generate new keys.**

    b. Enter a password for the keys.

    c. Click **Apply**.

    Only one set of keys is available at any one time. Therefore, when new keys are generated, the previous sets are no longer valid. The generated key should now be export to a file that can later be imported into other applications:

    a. Select **Export to keyfile**.

    b. Enter the desired keyfile path and filename.

    c. Click **Apply**.

    If a keyfile has been generated by another application such as WebSphere Application Server, this may be imported into Connection Manager:

    a. Select **Import to keyfile**.

    b. Enter the keyfile name and password. This is the password used when the original key was generated.

    c. Click **Apply**.

    > **Note:** Ensure the keys and passwords match at all of the LTPA end points. For security reasons, it is recommend to change the password at regular intervals.

6. Click **OK** to complete the LTPA/SSO setup for the selected authentication profile.

16.4.2 Verifying SSO using LTPA

Figure 16-9 displays the configuration used for SSO between Connection Manager and WebSphere Everyplace Access using HTTP Access Services and LTPA.

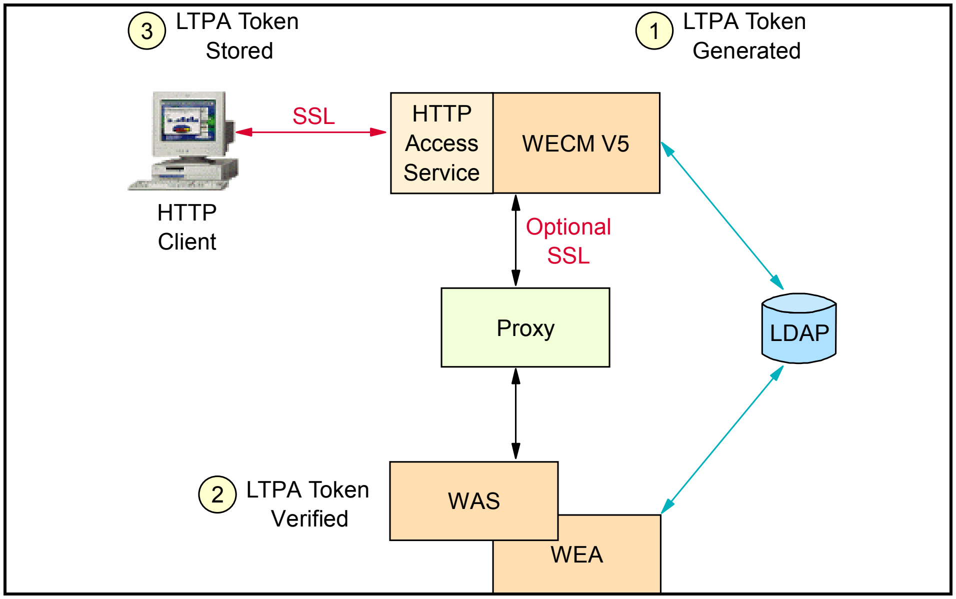


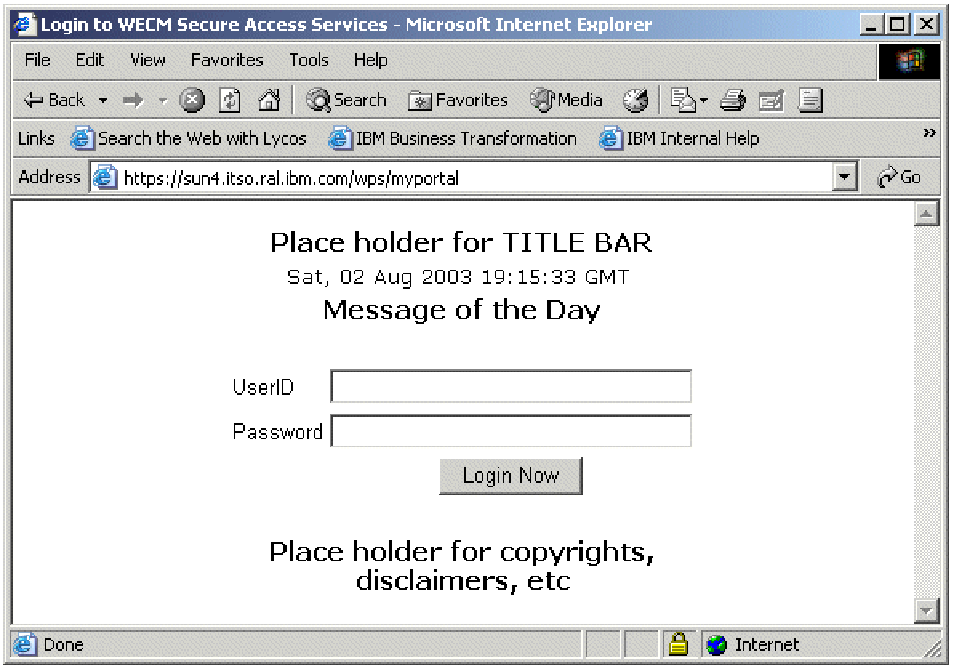
Figure 16-9 SSO example with LTPA

The proxy component shown in Figure 16-9 is optional. This is a configuration item set in the HTTP Access Service, and is not a dependency for SSO. However, many environments require the use of a proxy, hence the example. The proxy used here was the caching proxy component supplied with IBM Edge Server 2. Consideration must be given to the way in which the proxy is to be used, such as whether it will act as a forward or reverse proxy.

The LDAP has been shared between both Connection Manager and WebSphere Everyplace Access for the purpose of this example. SSO may also be achieved using separate LDAPs.

To confirm SSO to WebSphere Everyplace Access using HTTP services, complete the following steps:

1. From a Web browser, navigate to `http://<WECM_HTTP_ACCESS_URL>/wps/myportal`.
2. The connection between the HTTP client and Connection Manager is secured through SSL. After accepting the valid server certificate, a page similar to Figure 16-10 will be displayed. This page may be customized using the samples provided in <WECM_HOME>/http/msg/en. Enter the user details and click **Login Now**.



Login to WECM Secure Access Services - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Links
Search the Web with Lycos
IBM Business Transformation
IBM Internal Help
Address
Go
Done
Internet
Login Now

https://sun4.itso.ral.ibm.com/wps/myportal

Place holder for TITLE BAR

Sat, 02 Aug 2003 19:15:33 GMT

Message of the Day

UserID

Password

Place holder for copyrights, disclaimers, etc

Figure 16-10 Connection Manager HTTP access logon

3. Connection Manager generates an LTPA token, which is forwarded with the request to WebSphere Application Server. The user credentials are extracted from the token and used by WebSphere Everyplace Access to verify whether or not the user is a valid WebSphere Everyplace Access user. If so, the request should pass through to the user's WebSphere Everyplace Access home page without any need for the user to again enter the credentials.
4. The token will now be stored by the HTTP client until it expires or the user terminates the session. The user will not be required to reauthenticate to WebSphere Everyplace Access while the token is still valid.

If a suitable level of logging is enabled on the Connection Manager, the details similar to that shown in Example 16-5 will verify the LTPA token has been generated by Connection Manager. This can provide useful information for troubleshooting transactions. The extract has been taken from wg.log.

Example 16-5 LTPA entries in wg.log

```
1600:  108 (Aug 02 03/15:38:02.8976): AUTH_Server::genLtpaToken - created ltpa
token for user uid=test1,ou=sun4,dc=itso,dc=com on sso domain itso.ral.ibm.com
with TTL 240
  1600:  108 (Aug 02 03/15:38:03.9105): HTTP_Service::setupSession - ltpa token
set as the active key
```

```
Set-Cookie:
LtpaToken=qzt1IAY/zXlWbGS9m5RSz2AMoH3098B/7NIdZPyWmQeqQGYQC58Z3duKirb3NQQUgOwge
1hzQtCEMfwLstk4gtO/zeBODe3OTVcH6CUAM+/NpYuwGa5LFXgIMNnpNTtcXdCrwhOfNcApDkawSF5R
vbZqJo9T3XzrCJ67QWs/eJQ6VMBj9bVyTq5ji7SlVS61JWoTm/6+J/AsotmMQ7ykyPApsGAeNIELNKn
qOdh9+G7+YTUuSi8PerxdUVkZUTPYCuEcdSmLtebhCpWRP3ALUNw/3zAPcVu3cpTfjHZka8oTHnCY1N
+YffxIaUMhSu1lPJ5NNMpFgrmF7UGFzTQwCQqbTQ1i+rQJ;Domain=itso.ral.ibm.com
```

Similar information may also be seen in the trace files created by WebSphere Application Server. To generate this information, security tracing must be enabled. Consult the relevant WebSphere Application Server documentation for details on how to enable this. Example 16-6 shows an example of the corresponding request from Connection Manager as displayed in Example 16-5.

Example 16-6 Application server trace information

```
[8/1/03 15:43:43:375 EDT] 6d2fb95f WebAuthentica > getCookieValue
                                  LtpaToken
[8/1/03 15:43:43:375 EDT] 6d2fb95f WebAuthentica < getCookieValue

qzt1IAY/zXlWbGS9m5RSz2AMoH3098B/7NIdZPyWmQeqQGYQC58Z3duKirb3NQQUgOwge1hzQtCEMfw
Lstk4gtO/zeBODe3OTVcH6CUAM+/NpYuwGa5LFXgIMNnpNTtcXdCrwhOfNcApDkawSF5RvbZqJo9T3X
zrCJ67QWs/eJQ6VMBj9bVyTq5ji7SlVS61JWoTm/6+J/AsotmMQ7ykyPApsGAeNIELNKnqOdh9+G7+Y
TUuSi8PerxdUVkZUTPYCuEcdSmLtebhCpWRP3ALUNw/3zAPcVu3cpTfjHZka8oTHnCY1N+YffxIaUMh
Su1lPJ5NNMpFgrmF7UGFzTQwCQqbTQ1i+rQJ
[8/1/03 15:43:43:375 EDT] 6d2fb95f WebAuthentica D A cookie was received. The
name is LtpaToken and the value is
qzt1IAY/zXlWbGS9m5RSz2AMoH3098B/7NIdZPyWmQeqQGYQC58Z3duKirb3NQQUgOwge1hzQtCEMfw
Lstk4gtO/zeBODe3OTVcH6CUAM+/NpYuwGa5LFXgIMNnpNTtcXdCrwhOfNcApDkawSF5RvbZqJo9T3X
zrCJ67QWs/eJQ6VMBj9bVyTq5ji7SlVS61JWoTm/6+J/AsotmMQ7ykyPApsGAeNIELNKnqOdh9+G7+Y
TUuSi8PerxdUVkZUTPYCuEcdSmLtebhCpWRP3ALUNw/3zAPcVu3cpTfjHZka8oTHnCY1N+YffxIaUMh
Su1lPJ5NNMpFgrmF7UGFzTQwCQqbTQ1i+rQJ
[8/1/03 15:43:43:375 EDT] 6d2fb95f WebAuthentica D base64 ltpa token:

qzt1IAY/zXlWbGS9m5RSz2AMoH3098B/7NIdZPyWmQeqQGYQC58Z3duKirb3NQQUgOwge1hzQtCEMfw
Lstk4gtO/zeBODe3OTVcH6CUAM+/NpYuwGa5LFXgIMNnpNTtcXdCrwhOfNcApDkawSF5RvbZqJo9T3X
zrCJ67QWs/eJQ6VMBj9bVyTq5ji7SlVS61JWoTm/6+J/AsotmMQ7ykyPApsGAeNIELNKnqOdh9+G7+Y
TUuSi8PerxdUVkZUTPYCuEcdSmLtebhCpWRP3ALUNw/3zAPcVu3cpTfjHZka8oTHnCY1N+YffxIaUMh
Su1lPJ5NNMpFgrmF7UGFzTQwCQqbTQ1i+rQJ
[8/1/03 15:43:43:375 EDT] 6d2fb95f WebAuthentica D Validating the LTPA token
that was retrieved from the cookie.
[8/1/03 15:43:43:375 EDT] 6d2fb95f WebAuthentica > validate
[8/1/03 15:43:43:406 EDT] 6d2fb95f Authenticatio > extractCredentialAttributes
[8/1/03 15:43:43:406 EDT] 6d2fb95f Authenticatio D
publicName:rs600035.itso.ral.ibm.com:389/test1
[8/1/03 15:43:43:406 EDT] 6d2fb95f Authenticatio D
realm:rs600035.itso.ral.ibm.com:389;userName:test1
[8/1/03 15:43:43:406 EDT] 6d2fb95f Authenticatio D
accessId:user:rs600035.itso.ral.ibm.com:389/uid=test1, ou=sun4,dc=itso,dc=com
```

```
[8/1/03 15:43:43:406 EDT] 6d2fb95f WebAuthentica < validate
[8/1/03 15:43:43:406 EDT] 6d2fb95f WebAuthentica D The LTPA token was valid.
```

Most Web browsers have an option to allow viewing of LTPA tokens as they are processed. For example, in Microsoft Internet Explorer V6, select **Tools -> Internet options**, then select the **Privacy** tab. Click **Advanced** and check the options to enable a prompt before accepting any cookies. Now, when cookies such as ones containing an LTPA token are received, a dialog will be displayed asking the user whether or not they wish to accept the cookie, and it also gives the option to view the cookie. This is another useful resource for troubleshooting. Consult the documentation of your HTTP client for more details regarding cookie handling.

16.5 Enabling Single Sign-On on Domino

Domino can be configured to use the LTPA token generated by Connection Manager and WebSphere. When using HTTP access, the LTPA token is generated by Connection Manager. If the Mobility Client is used, LTPA tokens need to be generated by WebSphere, which is properly configured with the Connection Manager TAI plugin before connecting to Domino.

This section assumes that you are familiar with general Domino administration tasks. Setup steps are for Domino R5 but the configuration process is same with Version 6.

To configure SSO on Domino:

1. Open the directory of the used domain (usually names.nsf). Open **Server\Servers** view and click **Web..>Create Web SSO Configuration**.

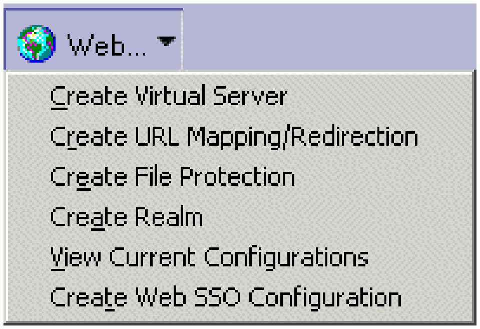
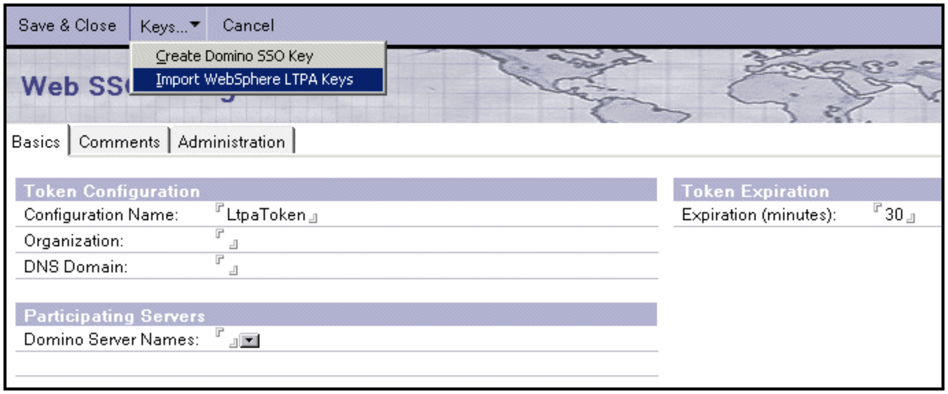

Web...
Create Virtual Server
Create URL Mapping/Redirection
Create File Protection
Create Realm
View Current Configurations
Create Web SSO Configuration

Figure 16-11 Create Web SSO configuration

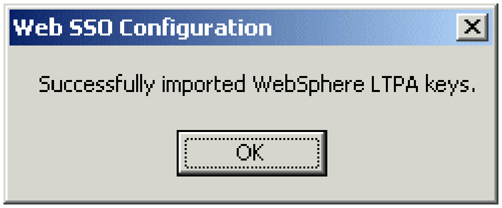
2. Select **Keys..>Import WebSphere LTPA Keys**.



Configuration Name: LtpaToken
Organization:
DNS Domain:
Domino Server Names:
Expiration (minutes): 30

Figure 16-12 Import WebSphere LTPA Keys

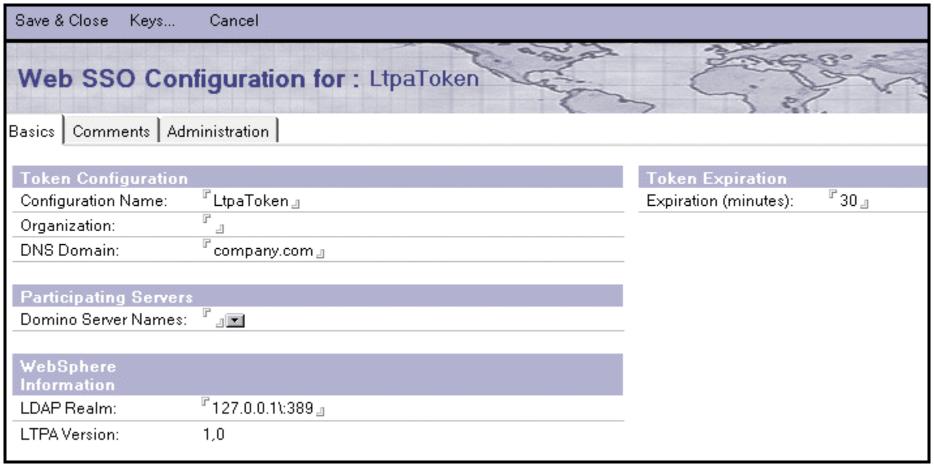
3. Enter the name of the exported LTPA key file and click **OK**. Enter the password for the imported LTPA. This is the same password you provided when you created your original LTPA keys, and click **OK**.
4. A confirmation message is displayed, indicating whether the import was successful or not. If it was not successful, repeat the steps.



Successfully imported WebSphere LTPA keys.

Figure 16-13 LTPA keys imported successfully

5. The used LDAP realm is filled in automatically; if there is also a port number provided, you must add a backslash \ before it in order for a LTPA token to be valid. Also, enter the names of all Domino servers on which you want to enable SSO.



Web SSO Configuration for : LtpaToken

Token Configuration

Configuration Name: LtpaToken

Organization:

DNS Domain: company.com

Token Expiration

Expiration (minutes): 30

Participating Servers

Domino Server Names:

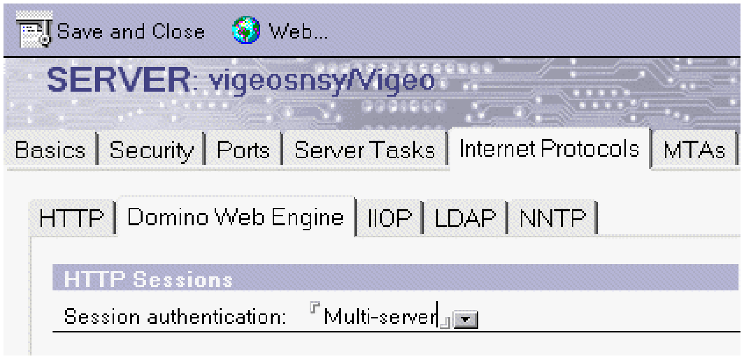
WebSphere Information

LDAP Realm: 127.0.0.1\:389

LTPA Version: 1.0

Figure 16-14 LDAP and Participation servers setup

6. Click **Save&Close** to complete the LTPA setup. You also need to enable a multi-server session authentication on each server using SSO.



SERVER: vigeosnsy/Vigeo

HTTP Sessions

Session authentication: Multi-server

Figure 16-15 Enable Session authentication

You may need to replicate the Domino directory to other servers in order for changes to take effect.

Restart \ the load HTTP task on Domino server. If everything is successful, the following message (or something similar) should appear in the console and log:

```
HTTP: Successfully loaded Web SSO Configuration.
```

If there are any error messages, first check that the server you are using is included in the `Participiants Servers` list in the *Web SSO Configuration* document.

You can now connect to Domino (after the LTPA is created by Connection Manager or WebSphere Application Server) using HTTP. You can also create an IIOP session from servlets and portlets using the LTPA token:

```
Session session = NotesFactory.createSession(<host>, <LtpaToken>);
```

Where *<host>* is the host name of the Domino server, and *<LtpaToken>* the LTPA token from the HTTP request.

16.6 Enabling Single Sign-On on Everyplace Access

For a sample scenario about Single Sign-On on WebSphere Everyplace Access, see Chapter 17, "WebSphere Everyplace Access Single Sign-On using LTPA" on page 347.

17

WebSphere Everyplace Access Single Sign-On using LTPA

This chapter provides information to establish Single Sign-On (SSO) between WebSphere Everyplace Connection Manager (WECM) and WebSphere Everyplace Access (WEA) using the Lightweight Third Party Authentication (LTPA). SSO provides a way of enhancing usability by reducing the number of authentication challenges presented to a user. So the only time a user has to enter its credentials is at the initial challenge provided by Connection Manager.

The topics discussed in this chapter include:

- Overview
- Sample scenario
- Connection Manager configuration
- WebSphere Everyplace Access configuration
- Running the scenario
- Troubleshooting

17.1 Overview

Many Connection Manager environments include at least one application server used to process and serve data to a range of clients. These applications may require the user to be authenticated prior to sending any requests. WebSphere Everyplace Access is an example of such an application whereby the user is presented with a challenge prior to being authenticated in the system.

In a standard Connection Manager environment such a challenge by an application means that the user has to re-enter their credentials despite already having done so to log in to the Connection Manager. For the sake of enhanced usability, it is recommended to try and avoid this type of double authentication.

This section discusses techniques that can be applied in a Connection Manager environment to establish a relationship between Connection Manager and downstream applications servers, thus creating single sign-on. With SSO, the user is only required to supply credentials for the initial Connection Manager login. Any following requests to appropriately configured application servers will recognize that the user has already been authenticated, and treats the requests accordingly.

Lightweight Third Party Authentication (LTPA)

Connection Manager is able to generate LTPA tokens that may be used as the mechanism for establishing SSO between Connection Manager and other supported applications for HTTP Access Services using HTTP access client authentication method. This is supported by both RADIUS and LDAP-bind type authentication profiles.

When enabled, an LTPA token is generated when a HTTP based user is authenticated by Connection Manager. This token is stored in a browser cookie to support SSO with other LTPA enabled application servers.

All servers using Connection Manager’s LTPA support must have the same LTPA keys and password, and reside in the same DNS domain. The key may be generated either through the Gatekeeper, or through a suitable application servers console such as WebSphere Everyplace Access.

17.2 Sample scenario

In this section, a sample scenario shown in Figure 17-1 is configured to illustrate the HTTP Access Services function implemented in Connection Manager. In this sample scenario, the following seven machines are used:

- Everyplace Connection Manager machine

- IBM AIX version V5.2.0.0 with Maintenance Package 02
- IBM DB2 Universal Database 7.2 with Fixpak 10
- IBM Directory Services V4.1
- IBM WebSphere Connection Manager V5.0.1.1

7. Reverse proxy server
   - Microsoft Windows 2000 Server
   - IBM WebSphere Edge Server V2.0
8. WebSphere Everyplace Access server
   - Microsoft Windows 2000 Server
   - IBM WebSphere Everyplace Access V4.3
9. Gatekeeper and HTTP client machines
   - Microsoft Windows 2000 Professional
   - IBM WebSphere Connection Manager Gatekeeper V5.0
   - Microsoft Internet Explorer V6.0

The sample scenario is illustrated in Figure 17-1.

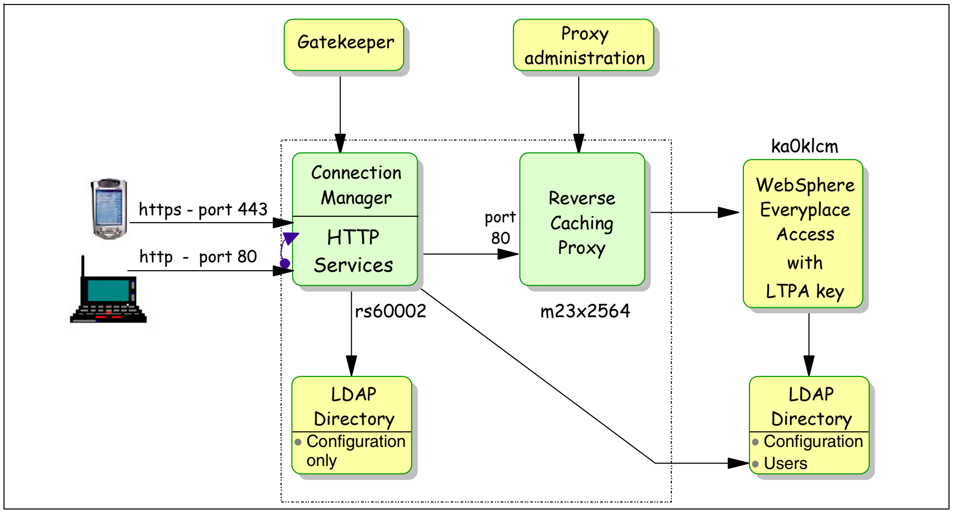


Figure 17-1 Single Sign-On using LTPA sample scenario

Connection Manager with HTTP Access Services

This machine runs HTTP Access Services in Connection Manager V5.0.1.1. For details about HTTP Access Services, see Chapter 14, “HTTP Access Services” on page 275.

Note: Connection Manager V5.0.1.1 or later is required for this scenario. It is recommended that you verify this requirement before implementing Single Sign-On with LTPA. For example, execute the following command to make sure this version is installed:

```
lswg -V
```

The result should be similar to the output shown in Example 17-1.

Example 17-1 Verifying Connection Manager version

```
IBM Websphere Everyplace Connection Manager Version 5.0.1.1 (5724-E80)
(C) COPYRIGHT International Business Machines Corp. and others 1994, 2004
Licensed Material -- Program Property of IBM --
All Rights Reserved

Feb  5 2004 / 20:47:37

Supported MNCs:
   sms-ois Short Message Service, OIS (TCP/IP)
```

Reverse proxy server

The reverse proxy machine plays a major role in the architecture, and it is responsible for redirecting the Everyplace Connection Manager messages to the WebSphere Everyplace Access server and vice-versa.

The reverse proxy must be specifically configured to work with Everyplace Connection Manager. Configuration options need to be made in the ibmproxy.conf file.

Note: For Windows machines this file is located in the C:\Program Files\IBM\edge\cp\etc\en_US directory, assuming that you are using default values when you installed IBM WebSphere Edge Server.

Port directive

In Chapter 15, “WebSphere Everyplace Access integration” on page 303, the HTTP Access Services scenario was configured to used port 8080. However, when using default configuration values, WebSphere Everyplace Access does not respond when using port 8080. For this sample scenario, port 80 is used because it is easier to change Connection Manager to use port 80 instead of configuring WebSphere Everyplace Access to use 8080.

Note: In previous product versions, Connection Manager did not accept port 80 as a valid port.

Therefore, the reverse proxy configuration file (ibmproxy.conf) port directive is used in this sample scenario and should look as shown in Example 17-2.

Example 17-2 Port directive

```
#        Port directive:
#        Port used by the server.
#        Default:  80
#        Syntax:   Port <num>
Port 80
```

SendRevProxyName directive

This directive must be configured so that application servers will also send HTTP traffic back to the reverse proxy. The configuration is illustrated in Example 17-3.

Example 17-3 SendRevProxyName

```
#        SendRevProxyName directive:
#
#        In a reverse proxy scenario, WTE normally sends the destination
#        origin server name in the HOST header of the request to the origin
#        server.  If this directive is set to yes, WTE will instead send
#        the WTE host name in the HOST header of the request to the origin
#        server.  This allows the origin server to use the WTE host name in
#        redirects sent back.  Therefore, subsequent requests to redirected
#        locations will go through WTE.
#
#        Default:  no
#        Syntax:   SendRevProxyName <yes | no>
#
# Example:
# SendRevProxyName no

SendRevProxyName yes
```

Mapping rules

The mapping rules need to be updated as they vary for different scenarios; this is because applications use different paths for their content location. The added directives for this sample scenario are illustrated in Example 17-4.

Example 17-4 Mapping rules

```
# ===================================================================== #
#        Mapping rules
# ===================================================================== #
Pass  /Admin/*.gif C:\PROGRA~1\IBM\edge\cp\server_root\Admin\en_US\*.gif
Pass  /Admin/*.html C:\PROGRA~1\IBM\edge\cp\server_root\Admin\en_US\*.html
# Pass  /* C:\PROGRA~1\IBM\edge\cp\server_root\pub\en_US\*
Pass  /pub/*     C:\PROGRA~1\IBM\edge\cp\server_root\pub\en_US\*

Proxy /          http://ka0klch.itso.ral.ibm.com/wps/myportal
Proxy /wps/*    http://ka0klch.itso.ral.ibm.com/wps/*
Proxy /*         http://ka0klch.itso.ral.ibm.com/*
```

Note: After updating the configuration, restart the caching proxy service.

WebSphere Everyplace Access server

This machine runs WebSphere Everyplace Access V4.3 and the installation for this scenario includes these components:

- WebSphere Everyplace Access
  - WebSphere Everyplace Access Core Services
- IBM Directory Server
- DB2 Universal Database Server

17.3 Connection Manager configuration

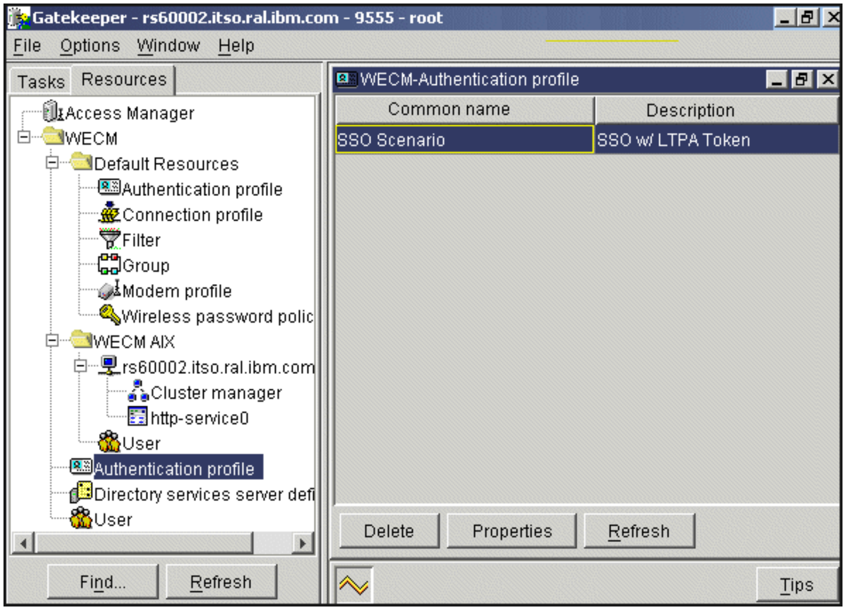
The Connection Manager configuration for this scenario will be done through Gatekeeper. The resources involved are:

- Authentication Profile
- HTTP Access Services
- Directory Services Server Definition
- Access Manager

17.3.1 Authentication profile

These are the steps:

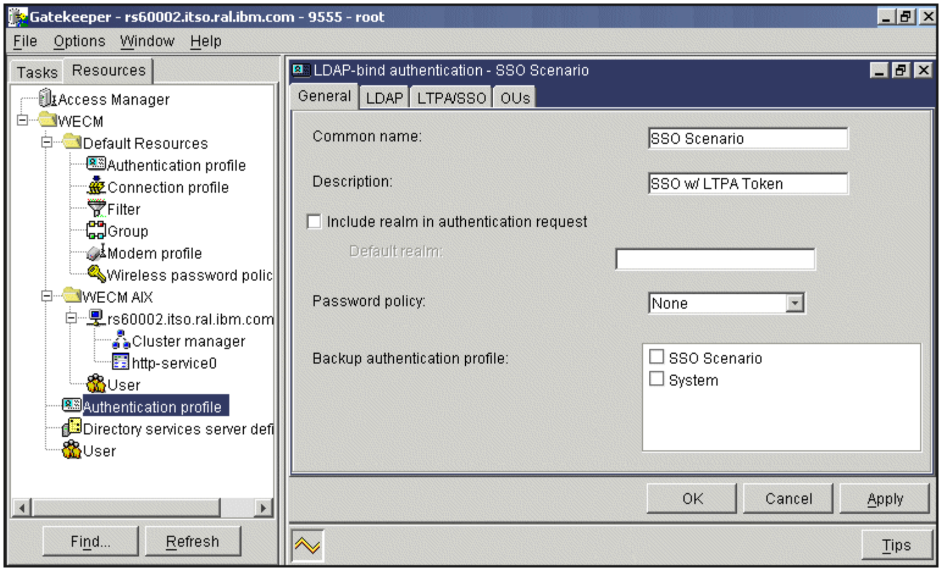
1. If it is not already created, create (add) an authentication profile resource.
2. After creating the authentication profile, double click the **Authentication Profile resource**. See Figure 17-2.



Common name	Description
SSO Scenario	SSO w/ LTPA Token

Figure 17-2 Authentication Profile List

3. Select **Authentication Profile Common Name** and click the **Properties** button.



Common name: SSO Scenario

Description: SSO w/ LTPA Token

□ Include realm in authentication request

Default realm:

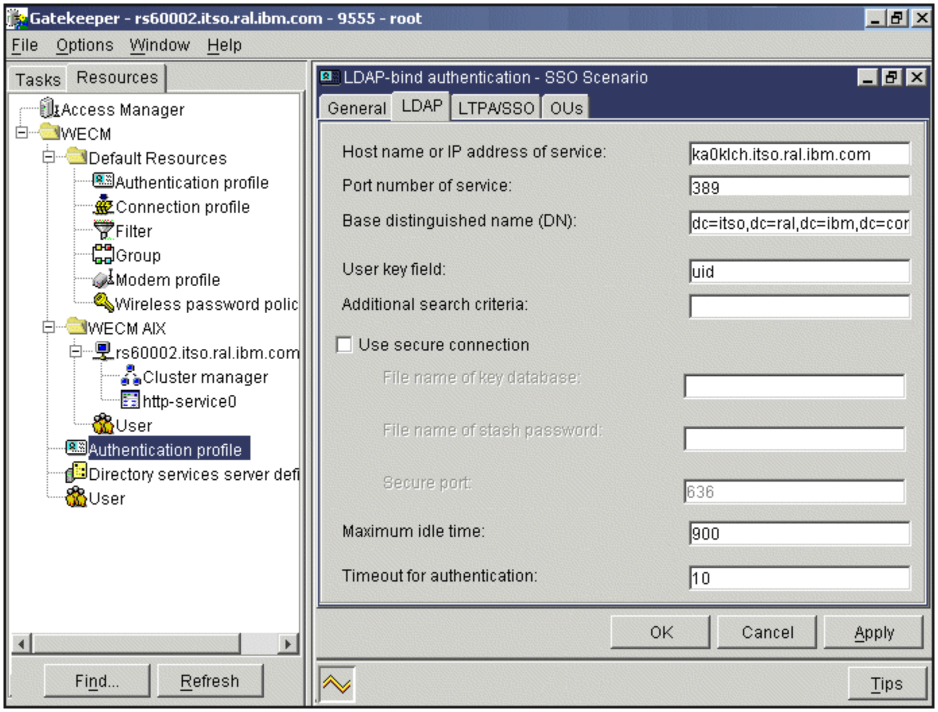
Password policy: None

Backup authentication profile: □ SSO Scenario □ System

Figure 17-3 Authentication Profile (General tab)

4. Select the **LDAP** tab. In the host name or IP address of service, we do not use the IP address but the fully qualified name of the WebSphere Everyplace Access machine. The generated LTPA token will contain this name, which is required to match when you import the key. Otherwise, the tokens will not match, and the scenario will not work properly.

   Note: An IP address can also be used in this field. However, you have to make sure that WebSphere Application Server is also set to use this IP address for authentication purposes. That is, the string has to match exactly.



Host name or IP address of service: ka0klch.itso.ral.ibm.com

Port number of service: 389

Base distinguished name (DN): dc=itso,dc=ral,dc=ibm,dc=cor

User key field: uid

Additional search criteria:

□ Use secure connection

File name of key database:

File name of stash password:

Secure port: 636

Maximum idle time: 900

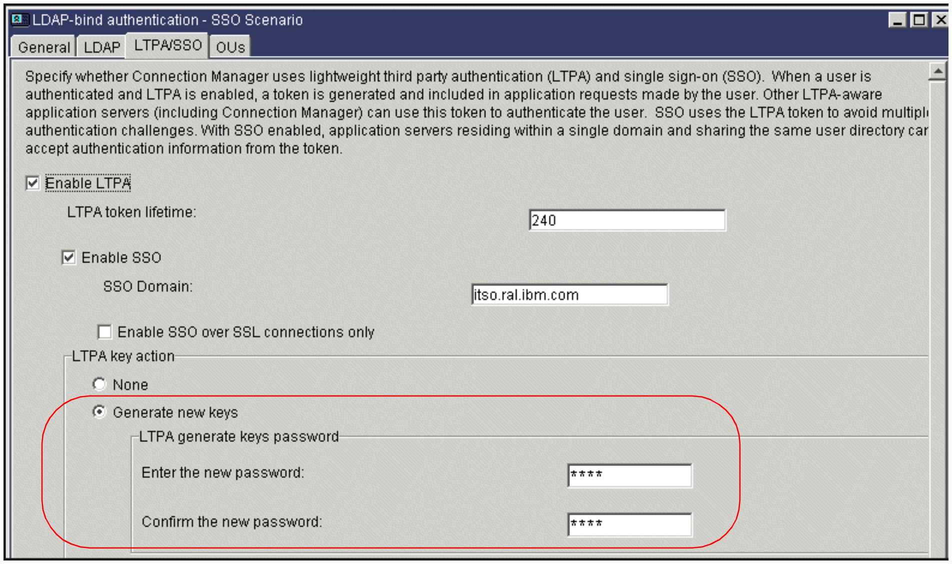
Timeout for authentication: 10

Figure 17-4 Authentication Profile (LDAP Tab)

5. Select the **LTPA/SSO** tab and generate a new LTPA key by checking the **Generate new keys** radio button, and then enter and confirm a password.

> **Note:** Keep and remember this password because it will be requested during the WebSphere Everyplace Access key import procedure.

The LTPA/SSO tab is shown in Figure 17-5.



Specify whether Connection Manager uses lightweight third party authentication (LTPA) and single sign-on (SSO). When a user is authenticated and LTPA is enabled, a token is generated and included in application requests made by the user. Other LTPA-aware application servers (including Connection Manager) can use this token to authenticate the user. SSO uses the LTPA token to avoid multipl authentication challenges. With SSO enabled, application servers residing within a single domain and sharing the same user directory car accept authentication information from the token.

☑ Enable LTPA

LTPA token lifetime: 240

☑ Enable SSO

SSO Domain: itso.ral.ibm.com

☐ Enable SSO over SSL connections only

LTPA key action

○ None

● Generate new keys

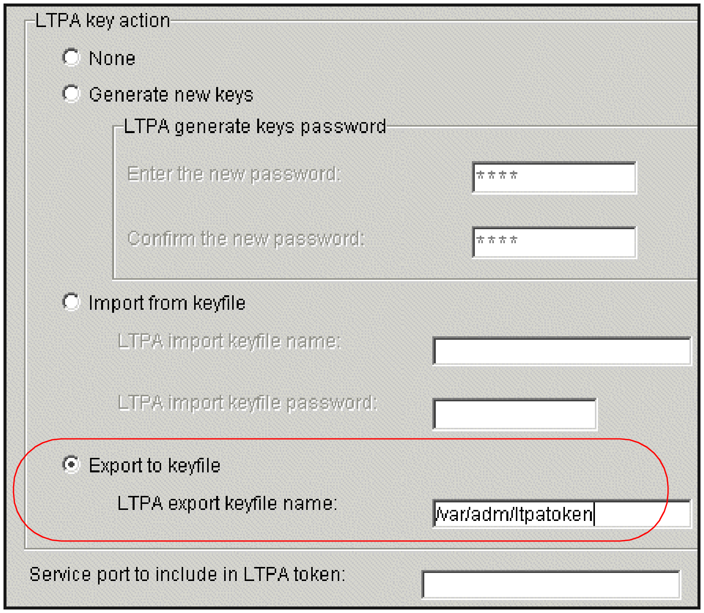
LTPA generate keys password

Enter the new password: ****

Confirm the new password: ****

Figure 17-5 Authentication profile - Generating new LTPA key

6. Once the key has been generated, it has to be exported to a file. The LTPA file will be created on /var/adm directory of the Connection Manager box.



LTPA key action

○ None

○ Generate new keys

LTPA generate keys password

Enter the new password: ****

Confirm the new password: ****

○ Import from keyfile

LTPA import keyfile name:

LTPA import keyfile password:

● Export to keyfile

LTPA export keyfile name: /var/adm/ltpatoken

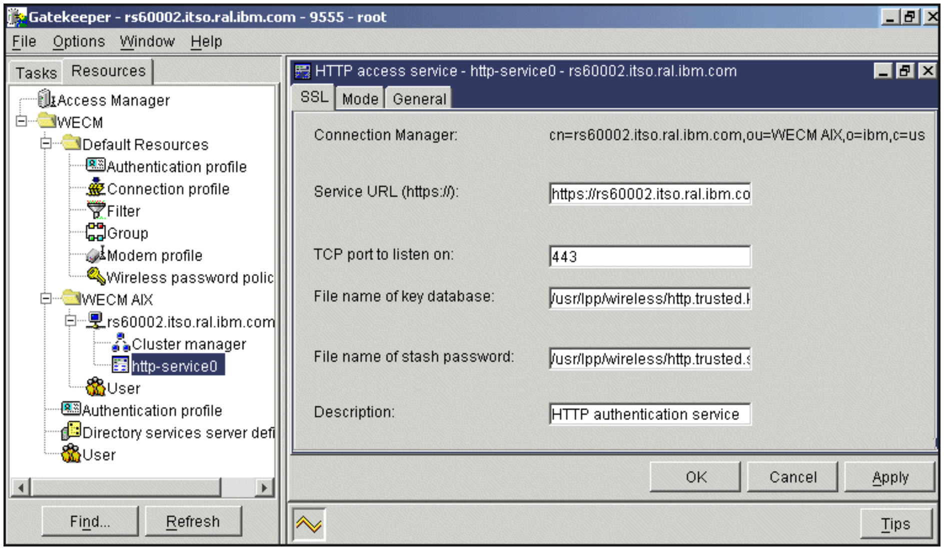
Service port to include in LTPA token:

Figure 17-6 Exporting LTPA key

17.3.2 HTTP Access Services

This section shows the HTTP Access Services configuration for the sample scenario where Connection Manager and WebSphere Everyplace Access are integrated to share LDAP users and implement Single Sign-On using LTPA. For example:

1. Log in to Gatekeeper, right-click and double click the **HTTP Access Service resource**.
2. The SSL tab is brought up as illustrated in Figure 17-7. In this window, all default fields are used.



Connection Manager: cn=rs60002.itso.ral.ibm.com,ou=WECM AIX,o=ibm,c=us

Service URL (https://): https://rs60002.itso.ral.ibm.co

TCP port to listen on: 443

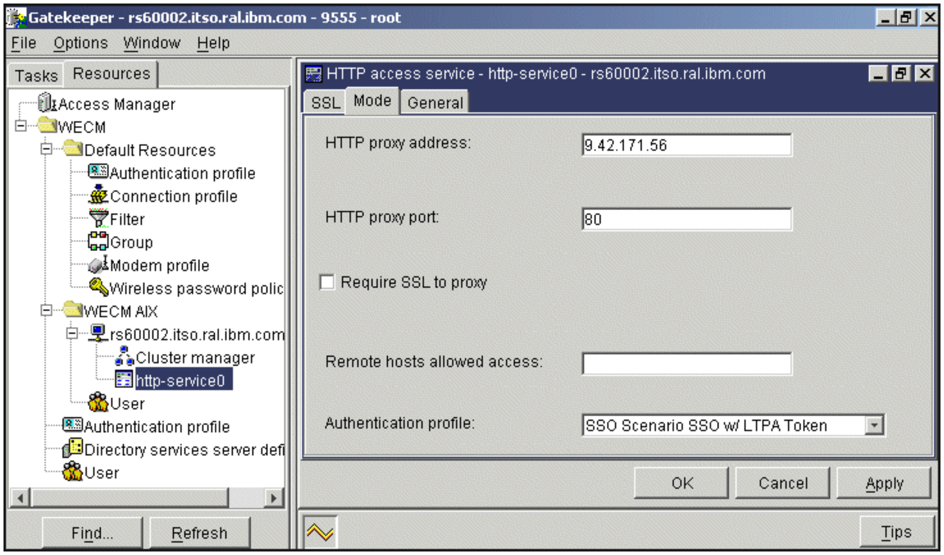
File name of key database: /usr/lpp/wireless/http.trusted.k

File name of stash password: /usr/lpp/wireless/http.trusted.s

Description: HTTP authentication service

Figure 17-7 HTTP Access Services (SSL tab)

3. Click the **Mode** tab and make sure the Authentication Profile field is set to use the SSO profile previously created.



HTTP proxy address: 9.42.171.56

HTTP proxy port: 80

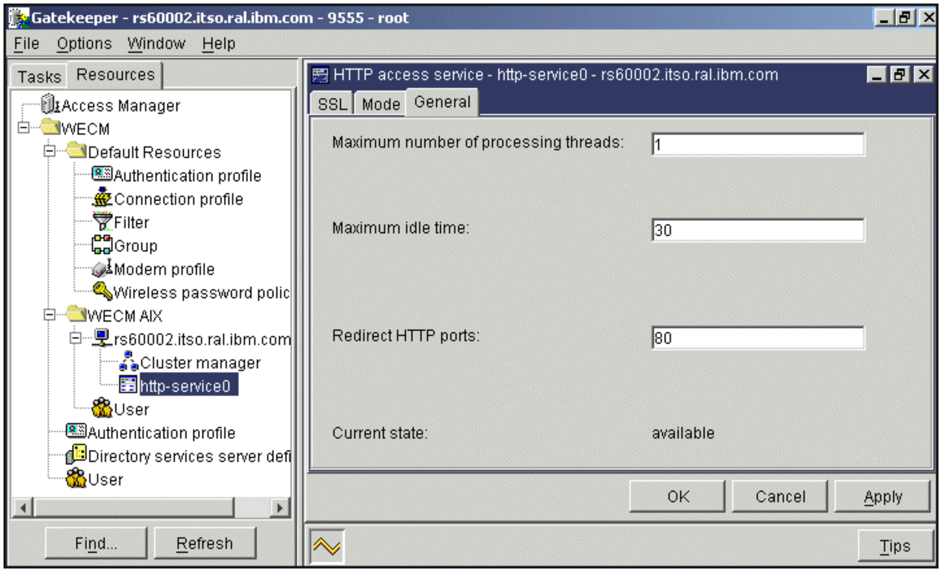
□ Require SSL to proxy

Remote hosts allowed access:

Authentication profile: SSO Scenario SSO w/ LTPA Token

Figure 17-8 HTTP Access Services (Mode tab)

4. Click the **General** tab and check that the Redirect HTTP ports field is set to 80 for this scenario.



Maximum number of processing threads: 1

Maximum idle time: 30

Redirect HTTP ports: 80

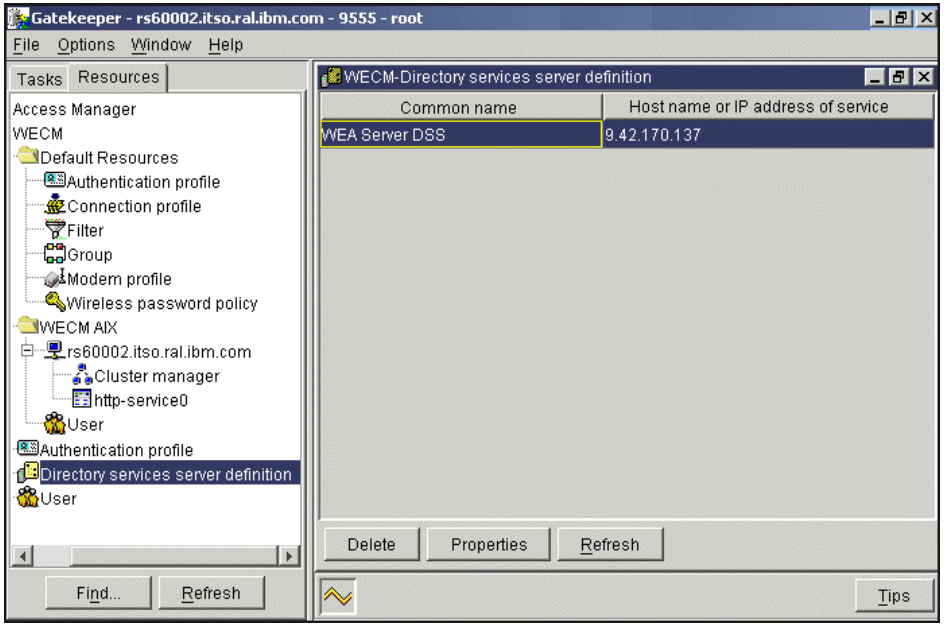
Current state: available

Figure 17-9 HTTP Access Services (General tab)

17.3.3 Directory Services Server

The LDAP directory for users is located in the WebSphere Everyplace Access box, and in order for Connection Manager to share the same directory this resource has to be created. For example:

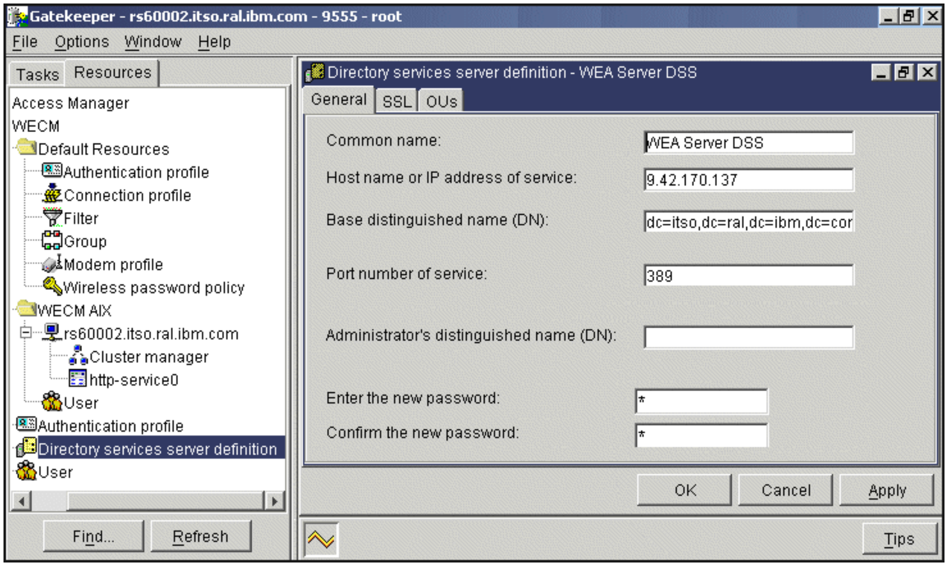
1. If it is not already created, add a Directory Services Server resource to Connection Manager.
2. Double click the **Directory Services Server resource**.



Common name	Host name or IP address of service
WEA Server DSS	9.42.170.137

Figure 17-10 Directory Services Server list

3. Select the resource created and click the **Properties** button; fill in the host name or IP address of service field with the LDAP server (in this scenario the WebSphere Everyplace Access box).



Common name: WEA Server DSS

Host name or IP address of service: 9.42.170.137

Base distinguished name (DN): dc=itso,dc=ral,dc=ibm,dc=cor

Port number of service: 389

Administrator's distinguished name (DN):

Enter the new password: *

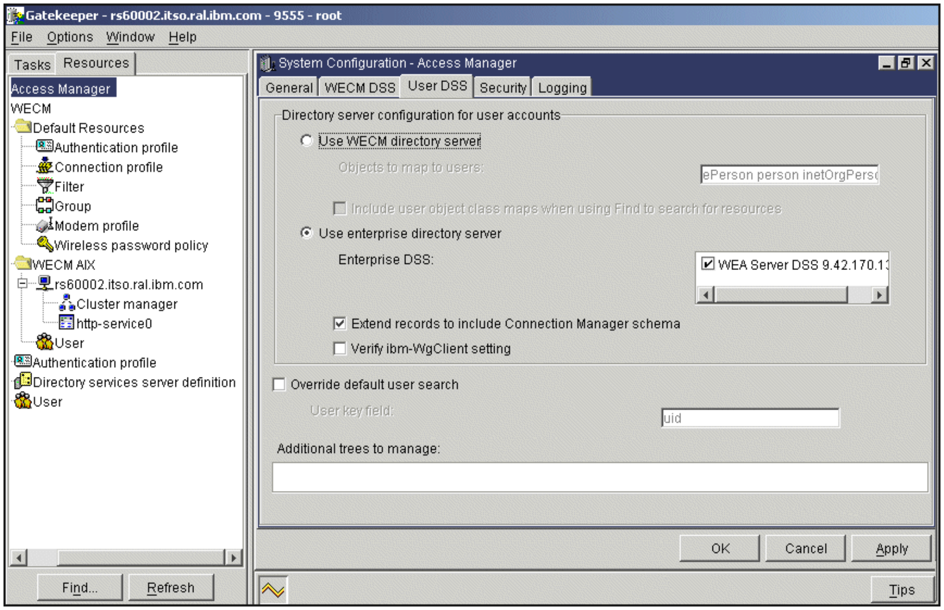
Confirm the new password: *

Figure 17-11 Directory Services Server (General tab)

17.3.4 Access Manager

Connection Manager has to point its requests for users to the newly created DSS. This is done by the Access Manager:

- Double click the **Access Manager resource** and click the **User DSS** tab. Select the user enterprise directory server, and check the box, which has the DSS that was created before.



Directory server configuration for user accounts

○ Use WECM directory server

Objects to map to users: ePerson person inetOrgPers(

□ Include user object class maps when using Find to search for resources

● Use enterprise directory server

Enterprise DSS: ☑ WEA Server DSS 9.42.170.1

☑ Extend records to include Connection Manager schema

□ Verify ibm-WgClient setting

□ Override default user search

User key field: uid

Additional trees to manage:

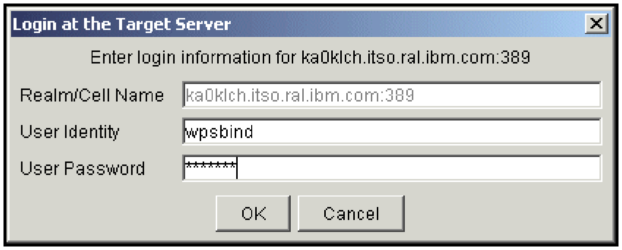
Figure 17-12 Access Manager (User DSS tab)

Note: At this time we recommend that you restart Connection Manager to make sure all the configuration updates are taken.

17.4 WebSphere Everyplace Access configuration

The exported LTPA key (see details in 17.3.1, “Authentication profile” on page 352) should now be imported into WebSphere Everyplace Access. For example, follow these steps to import the LTPA key:

1. Select **Start -> Programs -> IBM WebSphere -> Application Server 4.0 -> Administrator’s Console**.
2. As shown in Figure 17-13, log in to WebSphere Everyplace Access with the following parameters:
   – User: wpsbind
   – Password: wpsbind


Login at the Target Server

Enter login information for ka0klch.itso.ral.ibm.com:389

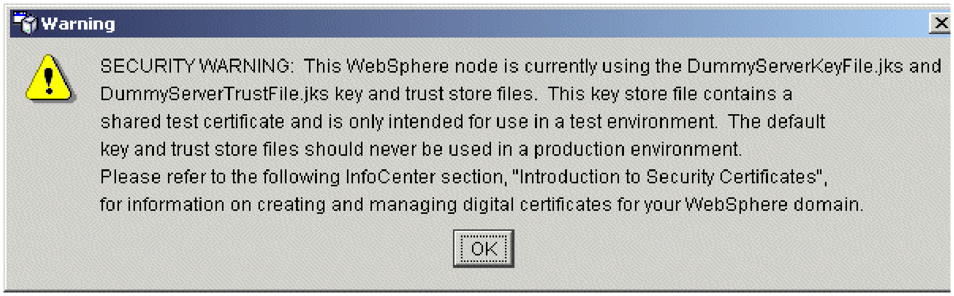
Realm/Cell Name: ka0klch.itso.ral.ibm.com:389

User Identity: wpsbind

User Password: *******

Figure 17-13 WebSphere Everyplace Access Login

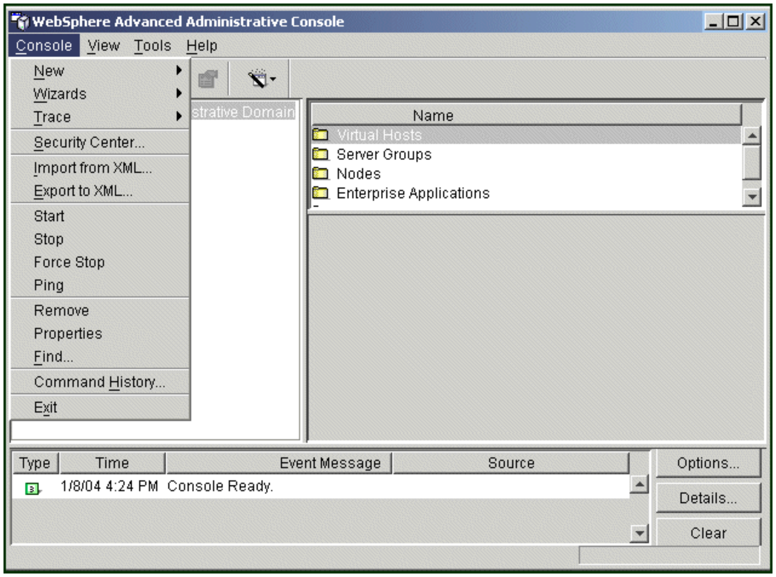
3. Since this is test environment, click **OK** to the warning window. However, for security reasons, you should never use the default and test certificates in a production environment.


Warning

SECURITY WARNING: This WebSphere node is currently using the DummyServerKeyFile.jks and DummyServerTrustFile.jks key and trust store files. This key store file contains a shared test certificate and is only intended for use in a test environment. The default key and trust store files should never be used in a production environment. Please refer to the following InfoCenter section, "Introduction to Security Certificates", for information on creating and managing digital certificates for your WebSphere domain.

Figure 17-14 Warning window

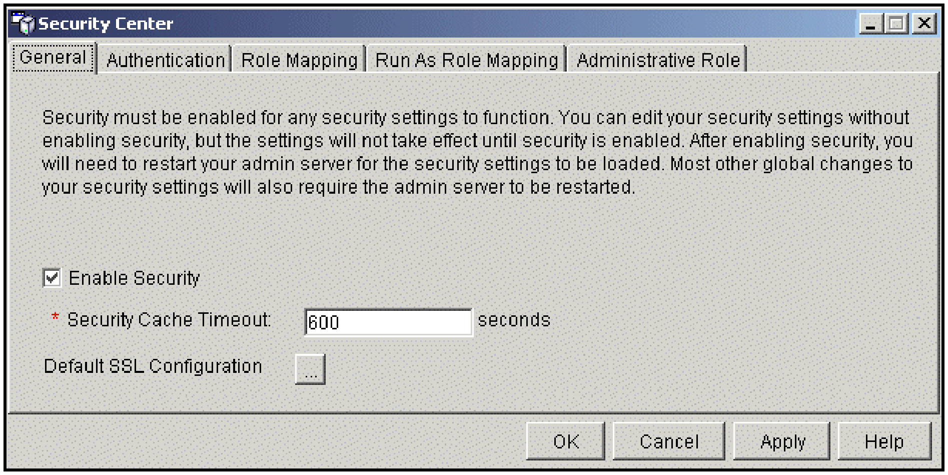
4. Open the **Security Center** as shown in Figure 17-15.



Type	Time	Event Message	Source
	1/8/04 4:24 PM	Console Ready.	

Figure 17-15 Security Center

5. Make sure the **Enable Security** check box is active on the General tab. See Figure 17-16.



Security must be enabled for any security settings to function. You can edit your security settings without enabling security, but the settings will not take effect until security is enabled. After enabling security, you will need to restart your admin server for the security settings to be loaded. Most other global changes to your security settings will also require the admin server to be restarted.

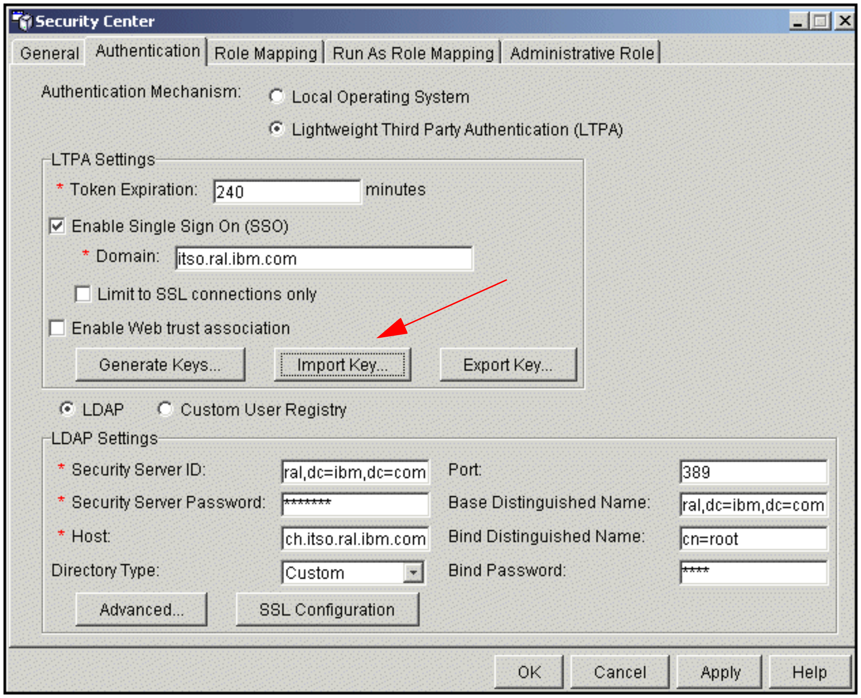
☑ Enable Security

* Security Cache Timeout: 600 seconds

Default SSL Configuration ...

Figure 17-16 Security Center (General tab)

6. Click the **Authentication** tab to import the LTPA key. See Figure 17-17.



Authentication Mechanism: ○ Local Operating System ● Lightweight Third Party Authentication (LTPA)

LTPA Settings

* Token Expiration: 240 minutes

☑ Enable Single Sign On (SSO)

* Domain: itso.ral.ibm.com

□ Limit to SSL connections only

□ Enable Web trust association

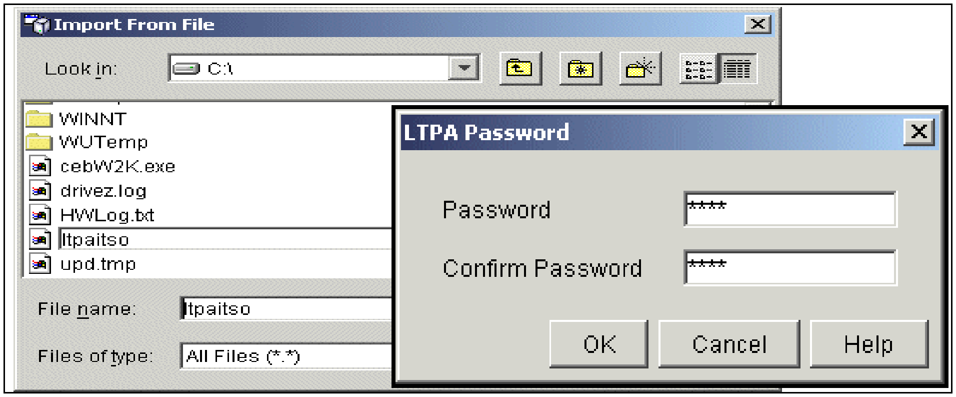
● LDAP ○ Custom User Registry

LDAP Settings

* Security Server ID:	ral,dc=ibm,dc=com	Port:	389
* Security Server Password:	*******	Base Distinguished Name:	ral,dc=ibm,dc=com
* Host:	ch.itso.ral.ibm.com	Bind Distinguished Name:	cn=root
Directory Type:	Custom	Bind Password:	****

Figure 17-17 Security Center (Authentication tab)

7. Import the generated LTPA key (exported from Connection Manager). Click the **Import Key** button, import the file, and type the same password you used when the LTPA key was created. Click **OK**.



File name: ltpaitso

Files of type: All Files (*.*)

Password ****

Confirm Password ****

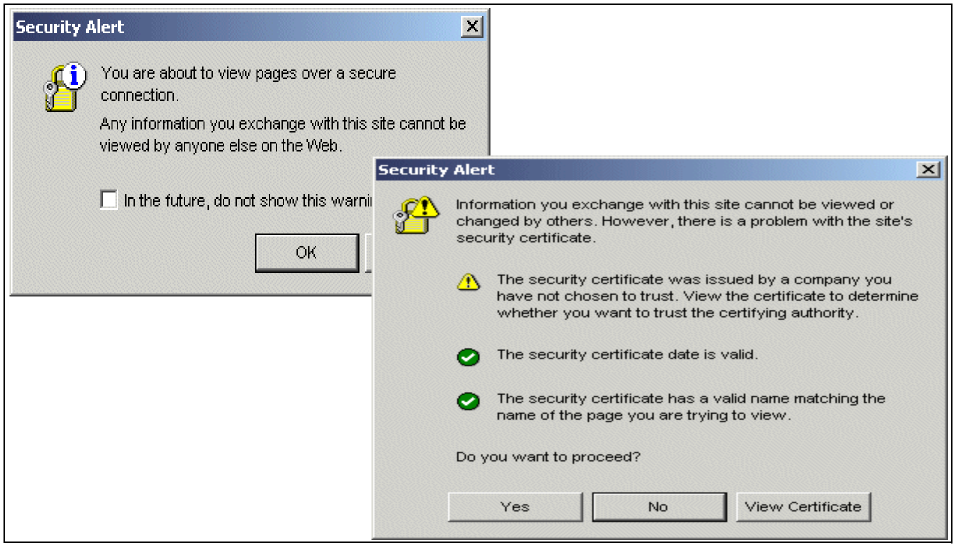
Figure 17-18 Import LTPA key from file

8. Click the **Apply** button on the Authentication tab (see Figure 17-17), then click **OK**.
9. Exit the Administrator's Console.
10. Stop and start the IBM WS AdminServer 4.0 service to make sure all the changes will be enabled.

17.5 Running the scenario

Once the proper configurations have been done, the scenario is ready to run. For example:

1. Open Internet Explorer.
2. Type the URL `https://rs60002.itso.ral.ibm.com/wps/myportal` and press the Enter key.
3. For testing purposes, click **OK** to any security alert windows.


Security Alert

You are about to view pages over a secure connection.

Any information you exchange with this site cannot be viewed by anyone else on the Web.

□ In the future, do not show this warni

Security Alert

Information you exchange with this site cannot be viewed or changed by others. However, there is a problem with the site's security certificate.

The security certificate was issued by a company you have not chosen to trust. View the certificate to determine whether you want to trust the certifying authority.

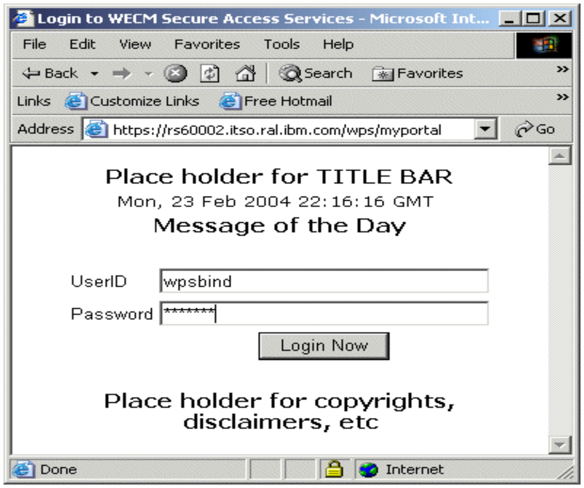
The security certificate date is valid.

The security certificate has a valid name matching the name of the page you are trying to view.

Do you want to proceed?

Figure 17-19 Security alert windows

4. Click **Yes** and type the Connection Manager user ID and password. For this scenario `wpsadmin` user is used; this user was created by default during the WebSphere Everyplace Access installation. Click **Login Now** to log in to Connection Manager.



Login to WECM Secure Access Services - Microsoft Int...
File Edit View Favorites Tools Help
Back
Search
Favorites
Links
Customize Links
Free Hotmail
Address
Go
Done
Internet

https://rs60002.itso.ral.ibm.com/wps/myportal

Place holder for TITLE BAR

Mon, 23 Feb 2004 22:16:16 GMT

Message of the Day

UserID wpsbind

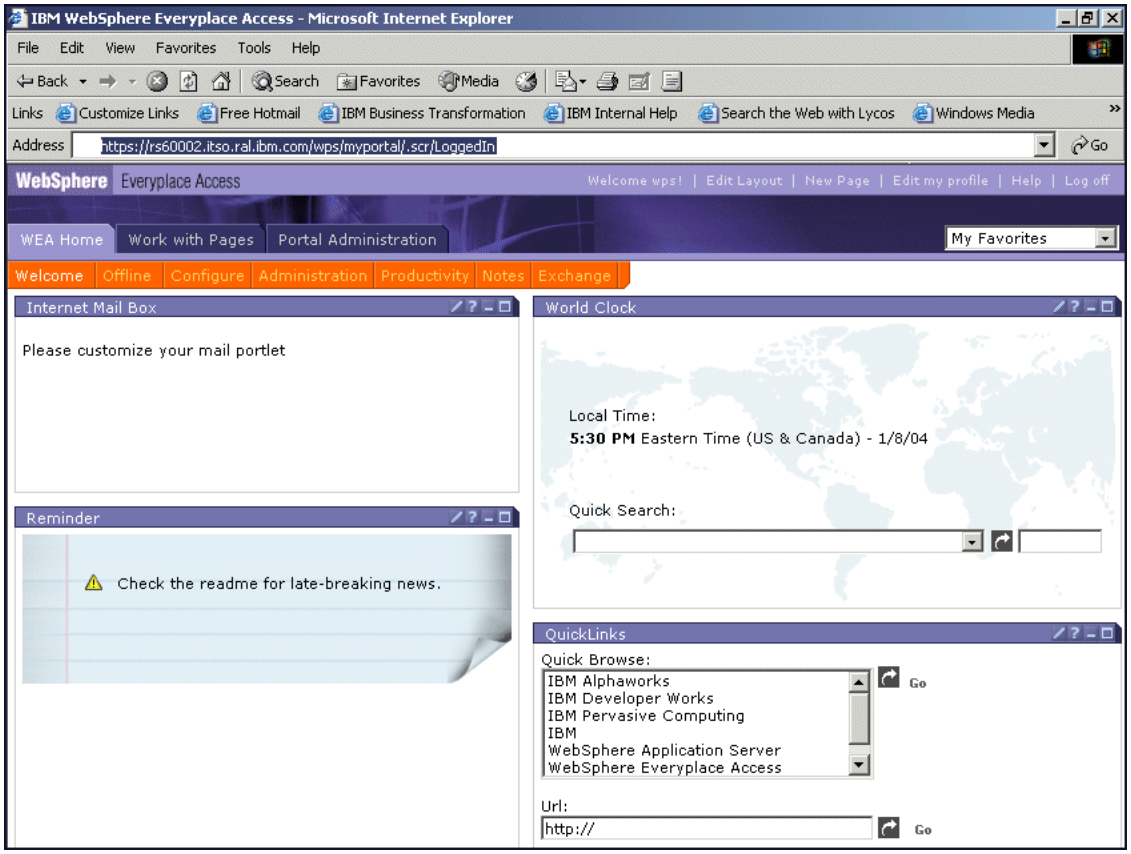
Password *******

Login Now

Place holder for copyrights, disclaimers, etc

Figure 17-20 Connection Manager login window

5. Connection Manager login activates the Single Sign-On function using LTPA, and the user is automatically authenticated by WebSphere Everyplace Access and connected to the portal page as shown in Figure 17-21.



Address: https://rs60002.itso.ral.ibm.com/wps/myportal/.scr/LoggedIn

Please customize your mail portlet

Local Time:
5:30 PM Eastern Time (US & Canada) - 1/8/04

Quick Search:

Check the readme for late-breaking news.

Quick Browse:
IBM Alphaworks
IBM Developer Works
IBM Pervasive Computing
IBM
WebSphere Application Server
WebSphere Everyplace Access

Url: http://

Figure 17-21 User portal page

17.6 Troubleshooting

This section provides some guidelines to help you troubleshoot potential problems you may encounter when implementing SSO with an LTPA token in Everyplace Connection Manager.

WebSphere Everyplace Access issues

In general, you will need to make sure that the applications run properly when not going through Everyplace Connection Manager. For example, for the sample scenario included in this chapter, the following URL is used to access applications directly from a desktop browser:

```
http://ka0klch.itso.ral.ibm.com/wps/myportal
```

Reverse proxy issues

It is recommended that you follow the request progress using the reverse proxy activity monitor function. For example, follow these steps:

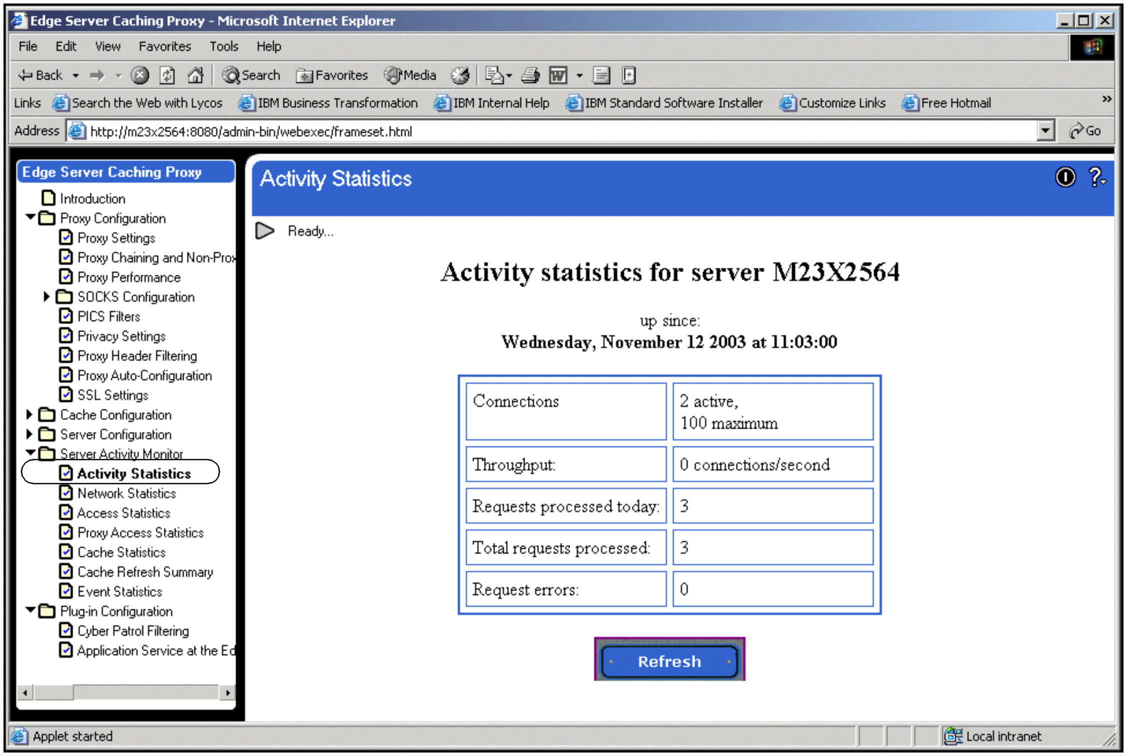
1. Access the WebSphere Edge Server Caching Proxy front page. Remember that the port directive has been changed in the configuration and uses the proper value. In the sample scenario included in this chapter, port 8080 is used. Therefore, to invoke the WebSphere Edge Server Caching Proxy front page the following URL should be used:

   ```
   http://m23x2564:8080/pub/FrntPage.html
   ```

2. Click **Configuration and Administration Forms** and log on to caching proxy administration.
3. Click **Server Activity Monitor -> Activity Statistics**.

Note: It is a good procedure to clear the activity statistics before running your scenario by refreshing the IBM Caching Proxy service.

Figure 17-22 illustrates the statistics for the reverse proxy traffic.



Address: http://m23x2564:8080/admin-bin/webexec/frameset.html

Ready...

Activity statistics for server M23X2564

up since:
Wednesday, November 12 2003 at 11:03:00

Connections	2 active, 100 maximum
Throughput:	0 connections/second
Requests processed today:	3
Total requests processed:	3
Request errors:	0

Figure 17-22 IBM WebSphere Edge Server activity statistics

SSO w/ LTPA transactions

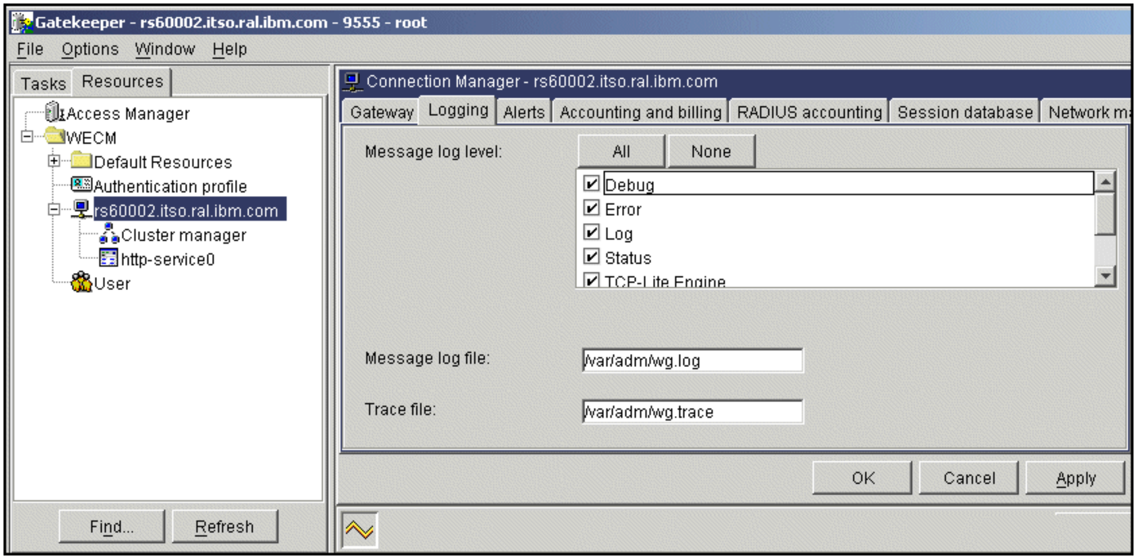
To make sure LTPA has been started properly, open the wg.log file. This file is located on /var/adm and look for LTPA. Table 17-5 shows a unsuccessful startup process.

Example 17-5 Unsuccessful startup)

```
cat wg.log
5602:  258 (Feb 23 04/18:46:23.3203):Enterprise DSS: failed to open data store
 (WEA Server DSS)
5602:  258 (Feb 23 04/18:47:38.3277):Enterprise DSS: failed to open data store
 (WEA Server DSS)
5602:  258 (Feb 23 04/18:48:53.3344):Enterprise DSS: failed to open data store
 (WEA Server DSS)
5602:  258 (Feb 23 04/18:50:08.3417):Enterprise DSS: failed to open data store
 (WEA Server DSS)
5602:  258 (Feb 23 04/18:51:31.8506):Enterprise DSS: failed to open data store
 (WEA Server DSS)
```

Everyplace Connection Manager logging

It is recommended to enable logging when troubleshooting SSO/LTPA problems in Everyplace Connection Manager. A sample configuration is shown in Figure 17-23 where logging levels are set, and the location of the log files are defined.



Gatekeeper - rs60002.itso.ral.ibm.com - 9555 - root
File Options Window Help
Tasks Resources
Access Manager
WECM
Default Resources
Authentication profile
rs60002.itso.ral.ibm.com
Cluster manager
http-service0
User
Find...
Refresh
Connection Manager - rs60002.itso.ral.ibm.com
Gateway Logging Alerts Accounting and billing RADIUS accounting Session database Network m
All
None
OK
Cancel
Apply

Message log level:
☑ Debug
☑ Error
☑ Log
☑ Status
☑ TCP-Lite Engine

Message log file: /var/adm/wg.log

Trace file: /var/adm/wg.trace

Figure 17-23 Logging configuration

Other miscellaneous recommendations

In general, it is always a good practice to do the following:

- Restart HTTP Access Services when its properties are changed.
- Restart Connection Manager whenever a LDAP change is made.
- Restart WebSphere Everyplace Access whenever a change on the security settings is made.

18

Messaging services

This chapter provides the understanding of enabling a Web application server to send messages to client devices such as a pager or a phone, which use a variety of wireless networks. Messaging services includes support for short message service (SMS) delivery, mobile-originated message delivery. When installed with the WAP proxy, messaging services also include support for unconfirmed Wireless Application Protocol (WAP) push delivery.

18.1 Messaging communications

Messaging services support several types of message modes:

- Short message (SMS)
- Short message delivery over proprietary networks (such as Mobitex or DataTAC)
- WAP push (when the WAP proxy is installed)
- E-mail using SMTP
- Simple network paging protocol (SNPP)

Examples of message operations might be news, stock quotes, pager messages, broadcast messages, and notification of events such as e-mail arrival.

18.1.1 Short message delivery operations

A short message delivery operation starts when a message processing application or servlets use the Connection Manager Messaging Services and *Push APIs* to send a message to the messaging services. The messaging services forward the message to a short message service center (SMS-C), an SMTP server, or other network server for subsequent delivery to a client.

Figure 18-1 shows a mobile device on the left receiving messages from messaging services, which receives the information from a messaging processing application.

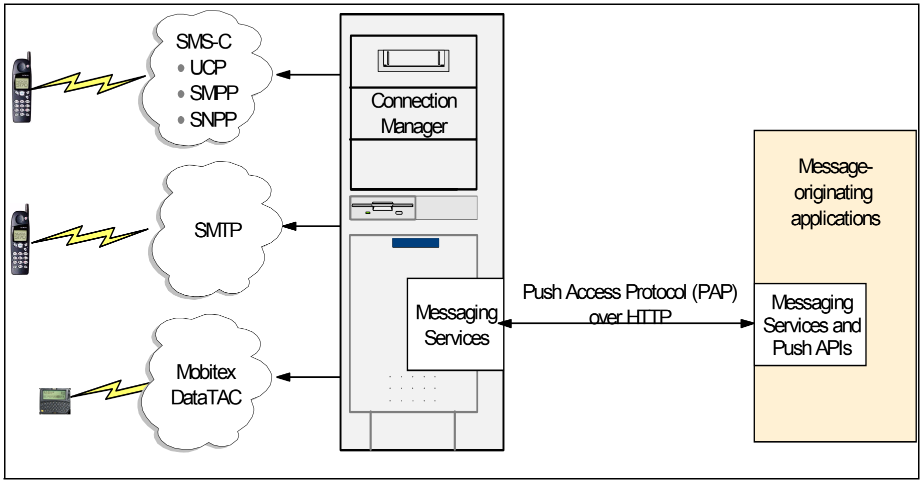

SMS-C
UCP
SMPP
SNPP
SMTP
Mobitex
DataTAC
Connection Manager
Messaging Services
Push Access Protocol (PAP) over HTTP
Message-originating applications
Messaging Services and Push APIs

Figure 18-1 Messaging communication

18.1.2 Mobile-originated message operation

Figure 18-2 shows a mobile device on the left sending messages to the messaging services, which forwards the information to a messaging processing application or servlet.

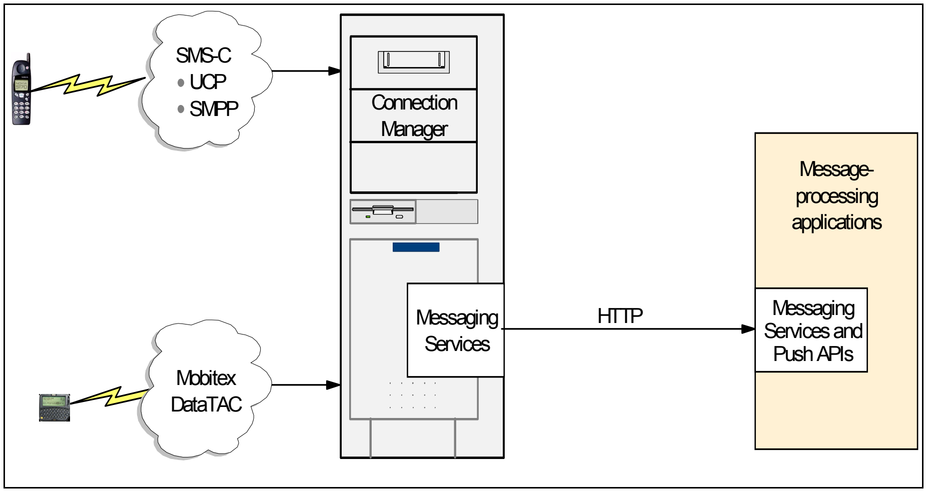


Figure 18-2 Messaging services accepting mobile-originated messages

A mobile-originated message operation starts when a client sends a message to a network provider for delivery to the messaging services. The messaging services use a HTTP post operation to forward the message to an application or servlet, which uses the Connection Manager Messaging Services and Push APIs.

18.2 Installation

This section assumes that the reader has previously read and understood bringing up the Gatekeeper and navigating through it. For details about the Connection Manager Gatekeeper, see Chapter 8, “Administration” on page 85.

After initializing WebSphere Everyplace Connection Manager Gatekeeper, select the **Resources** tab. Once you have selected the **Resource** tab (as illustrated in Figure 18-2) right-click the **Connection Manager system** to which you want to add the messaging services, then click **Add -> messaging services**.

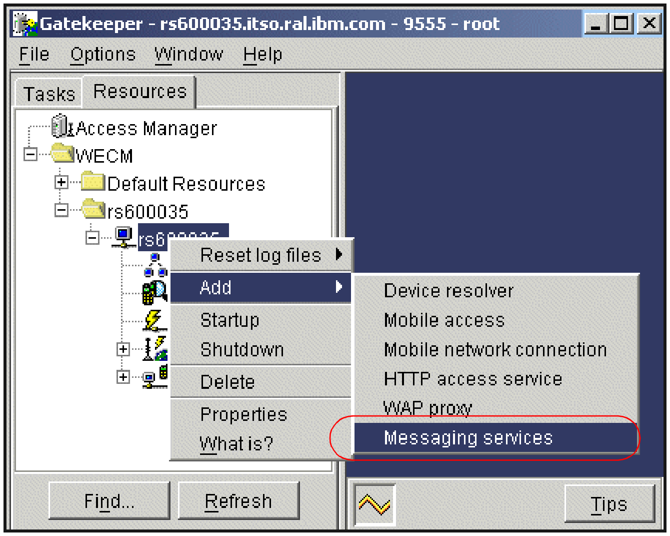
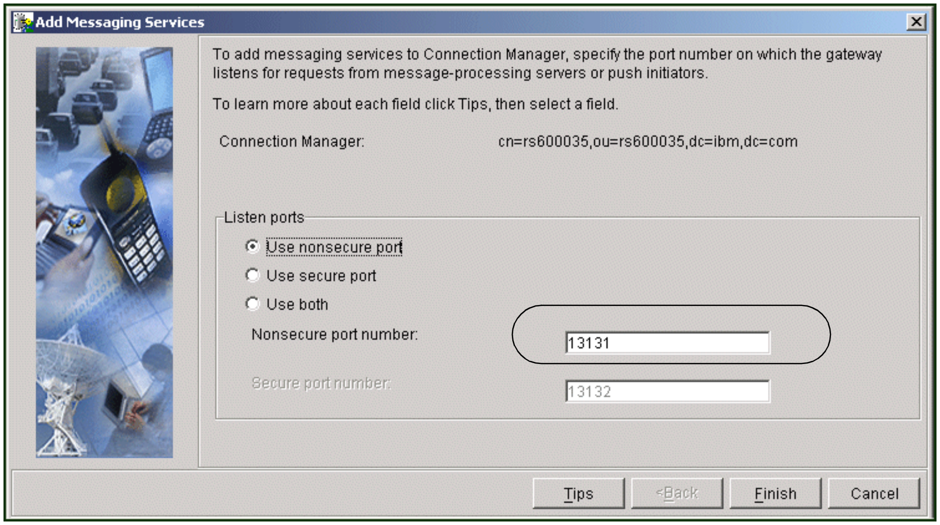


Figure 18-3 Adding Messaging services support

When a messaging service is defined to a Connection Manager system, the port number will be needed on which the Connection Manager listens for requests from applications using the Messaging Services and Push APIs. The default unsecure port is 13131, and the default secure port when SSL is implemented is 13132.



Add Messaging Services

To add messaging services to Connection Manager, specify the port number on which the gateway listens for requests from message-processing servers or push initiators.

To learn more about each field click Tips, then select a field.

Connection Manager: cn=rs600035,ou=rs600035,dc=ibm,dc=com

Listen ports

● Use nonsecure port
○ Use secure port
○ Use both

Nonsecure port number: 13131

Secure port number: 13132

Figure 18-4 Messaging services listening ports

After completion of listening port assignment, click the **Finish** button and note that Messaging Services has been added to you Resource tab as a resource for the Connection Manager you specified as shown in Figure 18-5.

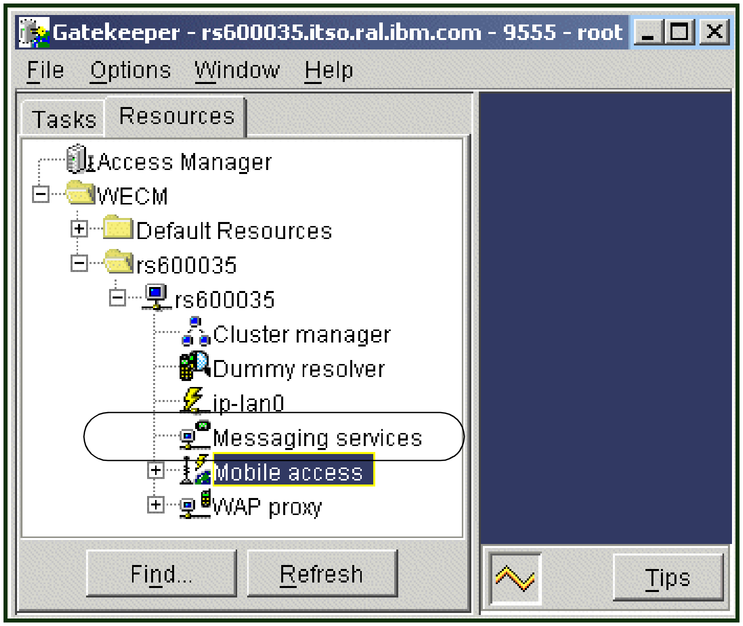


Figure 18-5 Messaging services

18.3 Messaging services configuration

This section will provide understanding the configuration of messaging services. Upon completion of adding the messaging services resource, you now need to define the resources used by the messaging services, enable secure communications for messaging services, and configure the MNC.

18.3.1 Application service

Used by the messaging services, an application service provides direct connection-based access to back-end server applications. An application service defines the protocol used between the application server and the messaging services. It identifies a port that is reserved for communication between an application on the messaging client device and the messaging services. The application provides a specific set of header information inside the data stream, which the messaging services uses to map the traffic to the back-end application server. There are two application service types: generic and pass-through.

Generic

The generic application service of messaging services provide traffic flow for applications between backend application servers and messaging services; the traffic flows over UDP, TCP, or TCP using SSL. Between the messaging client devices and the messaging services, the traffic flows over all MNCs except SMTP, SNPP, or dial-based MNCs. See Figure 18-33.

To configure a generic application service using the messaging service select the **Resource** tab of the Connection Manager you want to configure. Right-click **Messaging services**, and **Add application service ->Generic**.

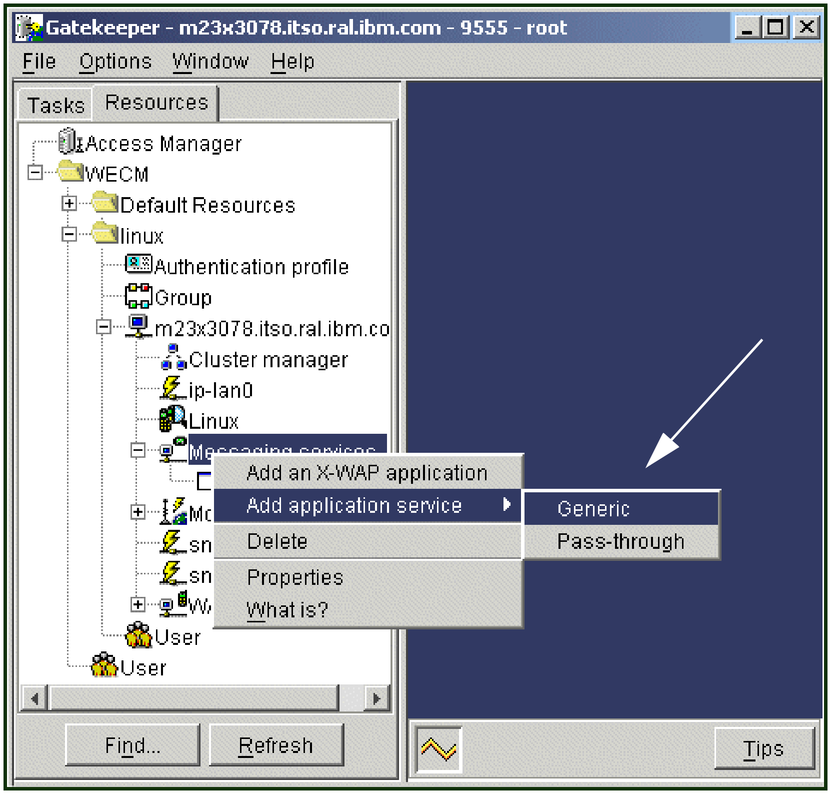
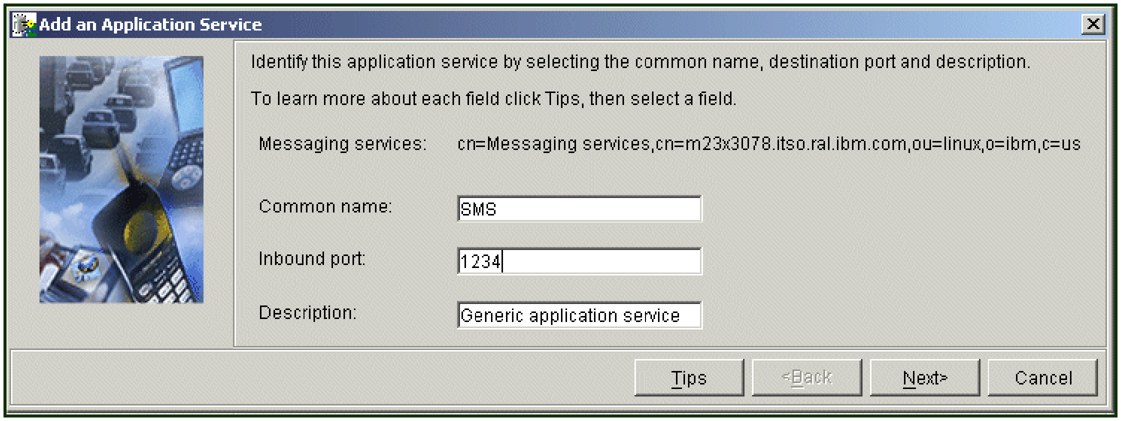


Figure 18-6 Adding a generic application service

As shown in Figure 18-7, the Add an application service screen will appear.



Identify this application service by selecting the common name, destination port and description.

To learn more about each field click Tips, then select a field.

Messaging services: cn=Messaging services,cn=m23x3078.itso.ral.ibm.com,ou=linux,o=ibm,c=us

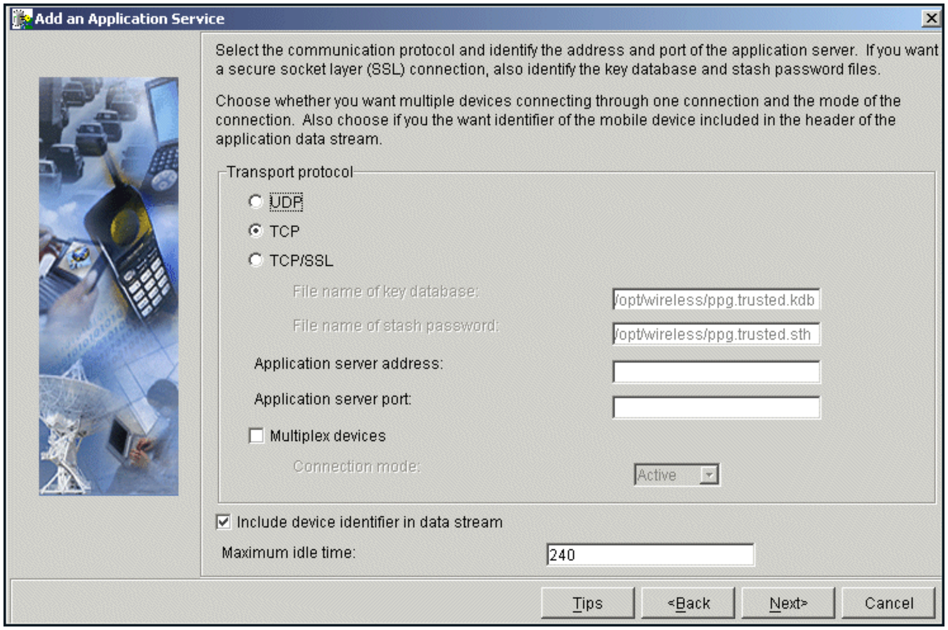
Common name: SMS

Inbound port: 1234

Description: Generic application service

Figure 18-7 Add an application service window

From the Add an application service screen, add a common name for the service, the inbound port that the data will come across, and the description. Select the **Next** button.



Add an Application Service

Select the communication protocol and identify the address and port of the application server. If you want a secure socket layer (SSL) connection, also identify the key database and stash password files.

Choose whether you want multiple devices connecting through one connection and the mode of the connection. Also choose if you the want identifier of the mobile device included in the header of the application data stream.

Transport protocol

○ UDP
● TCP
○ TCP/SSL

File name of key database: /opt/wireless/ppg.trusted.kdb
File name of stash password: /opt/wireless/ppg.trusted.sth
Application server address:
Application server port:

□ Multiplex devices

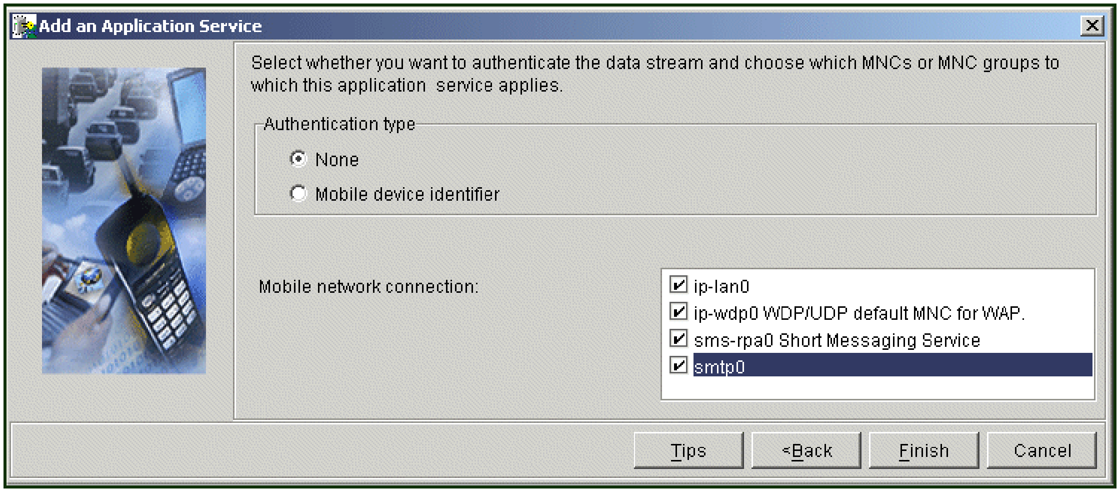
Connection mode: Active

☑ Include device identifier in data stream

Maximum idle time: 240

Figure 18-8 Add an Application Service

Next, select the communication protocol and identify the address and port of the application. Here you can also include the mobile identifier in the header of the application stream.



Select whether you want to authenticate the data stream and choose which MNCs or MNC groups to which this application service applies.

Authentication type

● None

○ Mobile device identifier

Mobile network connection:

☑ ip-lan0

☑ ip-wdp0 WDP/UDP default MNC for WAP.

☑ sms-rpa0 Short Messaging Service

☑ smtp0

Figure 18-9 Add an Application Service

Next, select whether you want to authenticate the data stream and choose which MNC groups to which the application service applies.

Pass-through

Pass-through application service provides traffic flow for applications between backend application servers and messaging client devices, but does not convert the protocol of the data stream. The messaging services keep accounting records of the data stream, which can be forwarded to another accounting server. Pass-through application services are supported on Mobitex and DataTAC-based MNCs.

Figure 18-10 illustrates an example of a pass-through application service using messaging services.

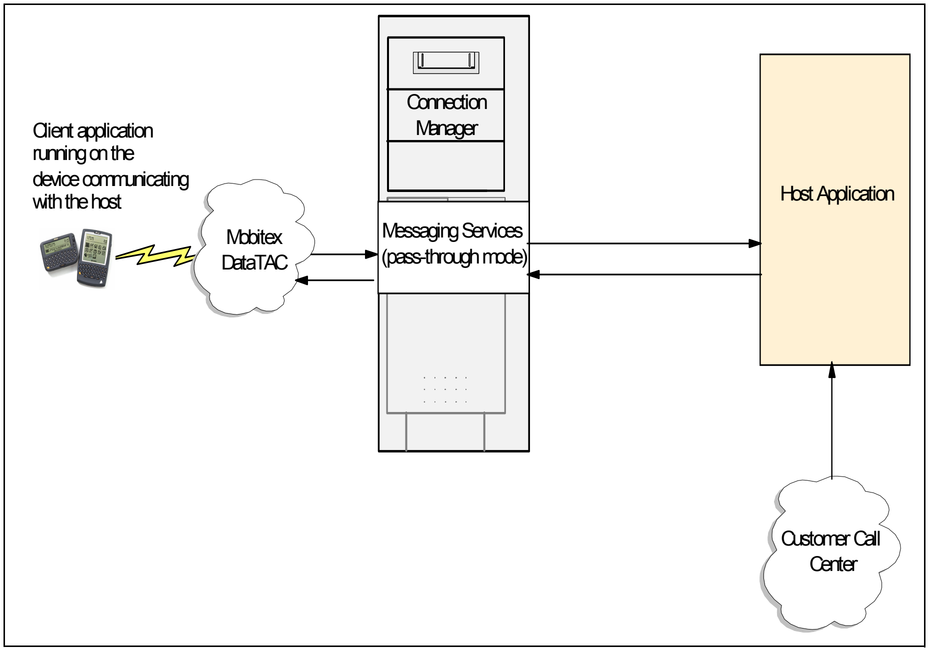

Client application running on the device communicating with the host
Mobitex DataTAC
Connection Manager
Messaging Services (pass-through mode)
Host Application
Customer Call Center

Figure 18-10 Pass-through mode

To configure a Pass-through application service using messaging service select the **Resource** tab of the Connection Manager you want to configure. Right-click **Messaging Services**, and click **Add application service -> Pass-through.**

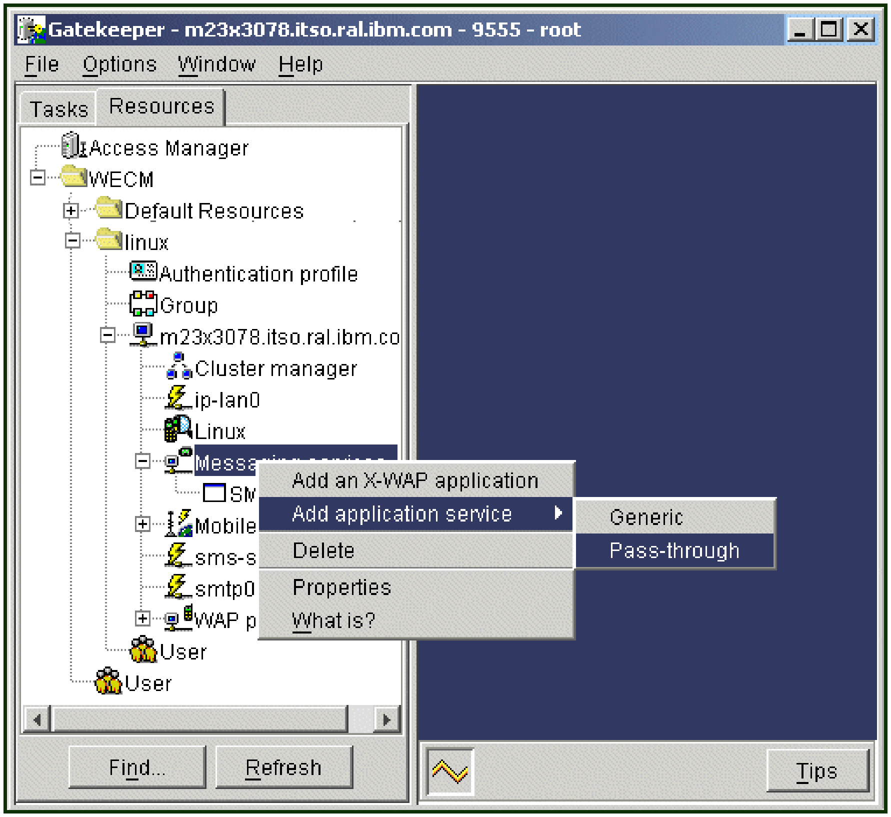
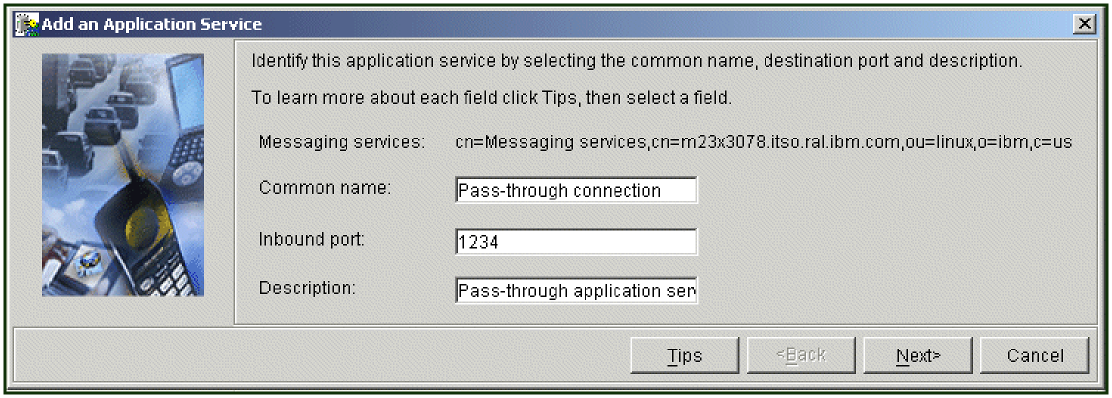


Figure 18-11 Pass-trough application service

The Add an application Service screen will appear.



Add an Application Service

Identify this application service by selecting the common name, destination port and description.

To learn more about each field click Tips, then select a field.

Messaging services: cn=Messaging services,cn=m23x3078.itso.ral.ibm.com,ou=linux,o=ibm,c=us

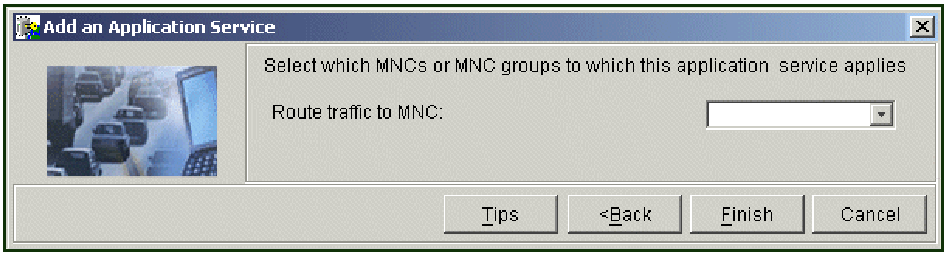
Common name: Pass-through connection

Inbound port: 1234

Description: Pass-through application ser

Figure 18-12 Adding an Application Service

From the Add an Application Service screen, add a common name for the service, the inbound port that the data will come across, and a description. Select the **Next** button.



Select which MNCs or MNC groups to which this application service applies

Route traffic to MNC:

Figure 18-13   Selecting MNCs or MNC groups

Next, select which MNCs or MNC groups to which this application service applies.

The last application service used by messaging services is the X-WAP application. When a push negotiator transmits content to a WAP client, it can target content to a specific client application. The application that is targeted on a WAP client is referred to as an X-WAP application. The push content can be delivered to any application as identified by the application ID on a WAP client. This identifier is a hex string that uniquely identifies the application on the client. This application ID is provided by the push initiator when the push content is submitted. When the client receives a push message, this application ID is used to identify the specified application on the WAP client.

The messaging services validates the incoming messages. If the content is accepted for delivery, messaging services deliver the message using Wireless Session Protocol (WSP) sessions to a WAP client.

To configure a X-WAP application service using messaging service, select the **Resource** tab of the Connection Manager you want to configure. Right-click **Messaging Services**, and select **Add an X-WAP application**.

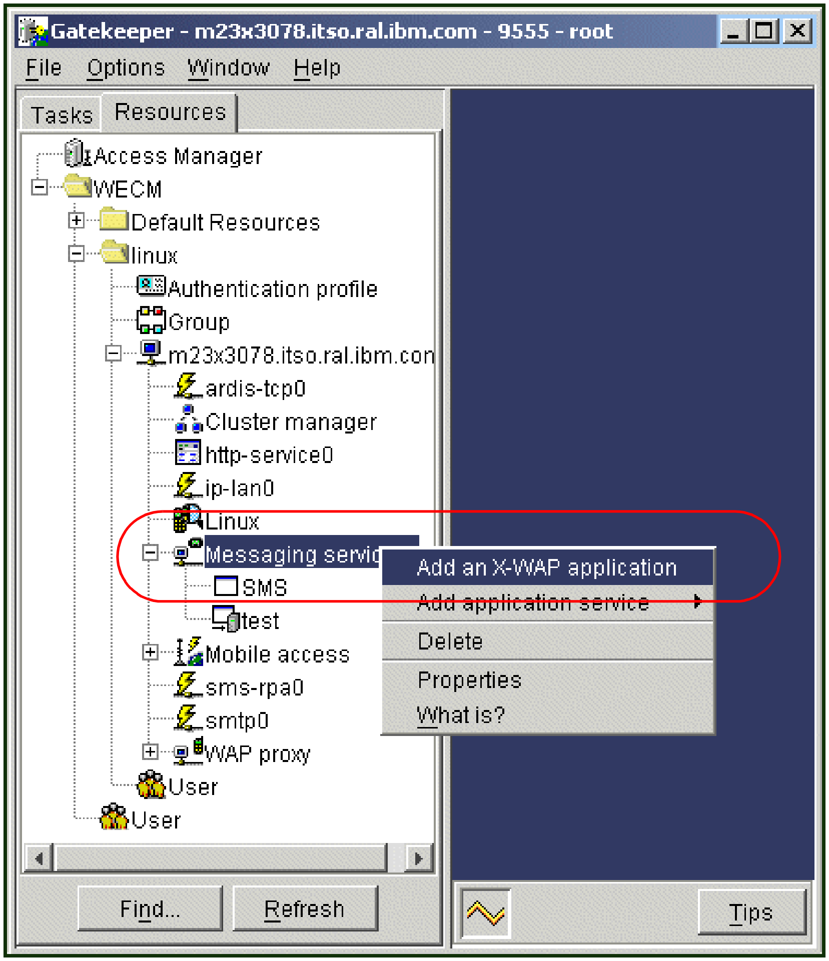
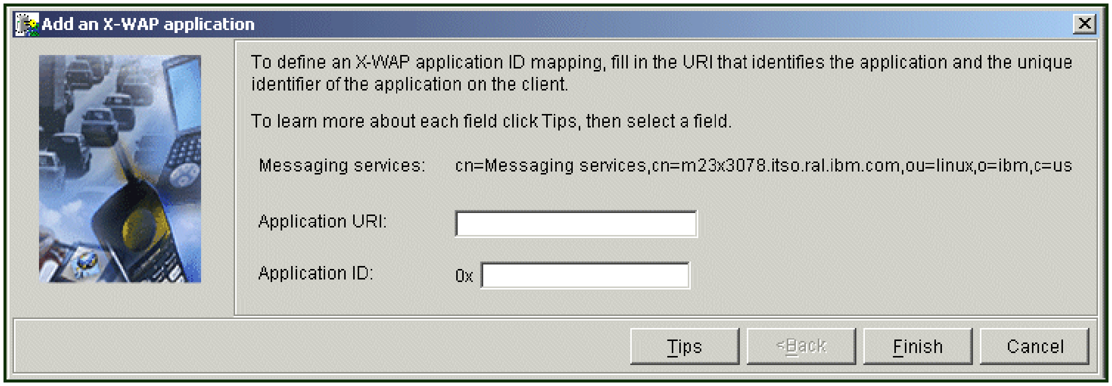


Figure 18-14 Add an X-WAP application

From the Add an X-WAP application service screen, define an X-WAP application ID mapping by filling in the URL that identities the unique identifier of the application on the client.



To define an X-WAP application ID mapping, fill in the URI that identifies the application and the unique identifier of the application on the client.

To learn more about each field click Tips, then select a field.

Messaging services: cn=Messaging services,cn=m23x3078.itso.ral.ibm.com,ou=linux,o=ibm,c=us

Application URI:

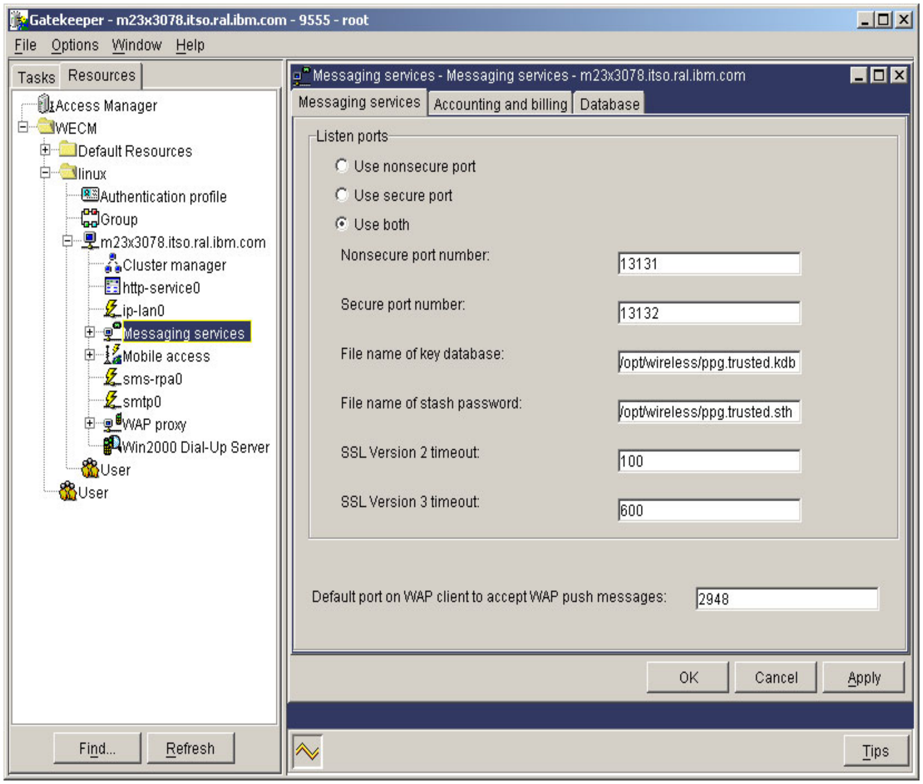
Application ID: 0x

Figure 18-15 Define X-WAP application ID mapping

18.3.2 Enabling secure communications for messaging services

You will need to configure SSL certificates for messaging services. To configure a secure connection between the messaging services and message processing services or push initiators, you must configure each endpoint. For example:

- The key certificates at the messaging services endpoint are stored in the key database and secured by a stash password.
- The default key database filename is ppg.trusted.kdb.
- The default stash password is “trusted” and is stored in ppg.trusted.sth.
- To change the key database filename or stash password filename, you have to edit the Gateway tab of the messaging services properties.



Listen ports

○ Use nonsecure port

○ Use secure port

● Use both

Nonsecure port number: 13131

Secure port number: 13132

File name of key database: /opt/wireless/ppg.trusted.kdb

File name of stash password: /opt/wireless/ppg.trusted.sth

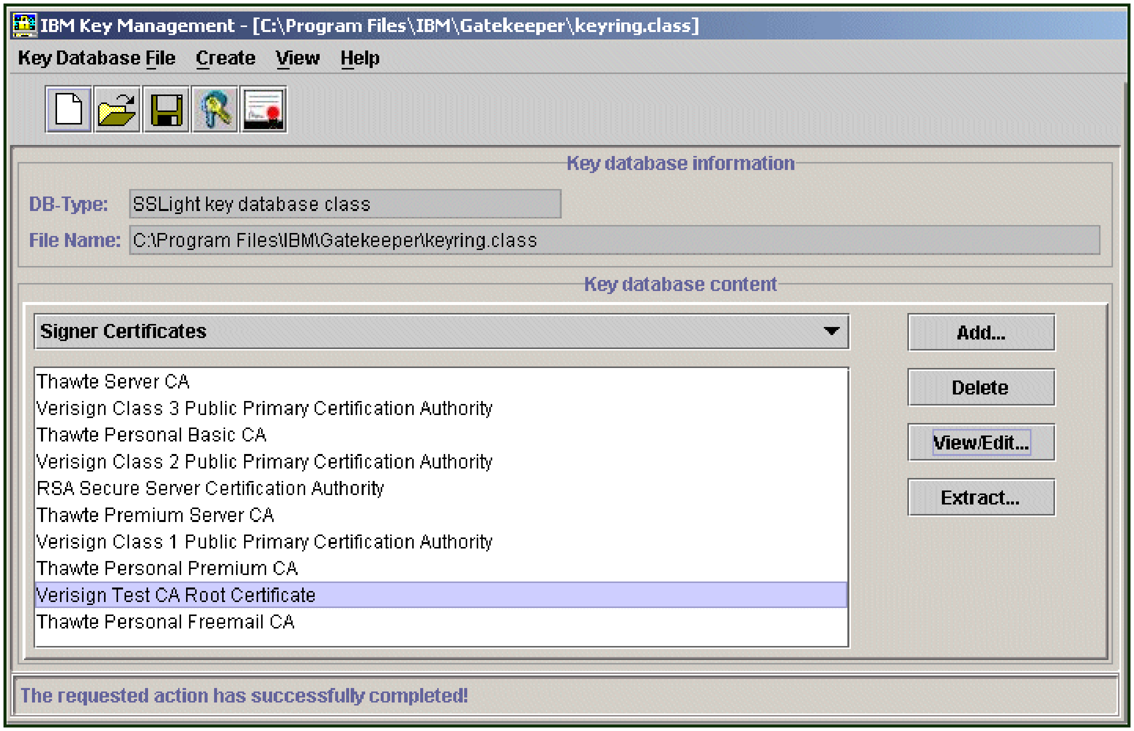
SSL Version 2 timeout: 100

SSL Version 3 timeout: 600

Default port on WAP client to accept WAP push messages: 2948

Figure 18-16 Messaging services

To add or view the certificates in the key database, you can select the IBM Key Management tool that was install with the Connection Manager software. Once you invoke the management tool you should see the following screen in Figure 18-16.



IBM Key Management - [C:\Program Files\IBM\Gatekeeper\keyring.class]

Key database information

DB-Type: SSLight key database class

File Name: C:\Program Files\IBM\Gatekeeper\keyring.class

Key database content

Signer Certificates

Thawte Server CA
Verisign Class 3 Public Primary Certification Authority
Thawte Personal Basic CA
Verisign Class 2 Public Primary Certification Authority
RSA Secure Server Certification Authority
Thawte Premium Server CA
Verisign Class 1 Public Primary Certification Authority
Thawte Personal Premium CA
Verisign Test CA Root Certificate
Thawte Personal Freemail CA

The requested action has successfully completed!

Figure 18-17 IBM Key Management

Note: To add or view key certificates in the key database on the server, log in as root and use the IBM Key Management interface by entering `wg_ikeyman` from the command line prompt.

For additional information about changing the stash password, creating a certificate request, submitting the certificate request, storing certificates, and updating the root certificate, click **Help ->Contents** using the IBM Key Management interface.

After making any changes to the key database, shut down then start up the Connection Manager to activate any changes.

SSL certificates for message processing applications

To configure a secure connection between the messaging services and applications using the Messaging Services and Push APIs, you must configure each endpoint.

To create a secure connection for applications using the Messaging Services and push APIs, you will need to:

- Create a self-signed certificate or issue a certificate request to obtain a certificate from a Certificate Authority.
- Add the certificate to the existing messaging service key databases, then extract the certificate.
- Create a new key database for use by the message-processing application and add the extracted certificate to it.system.
- Transfer the key database to the message-processing application's system.

> **Note:** See the Connection Manager Administration Guide v5 for addition information on the following:
>
> - Configuring key certificates from the messaging services key database
> - Creating a key database class file for Java applications
> - Creating a key database and stash file for C applications

Configuring an SMPP MNC for messaging services

A Short Message Peer-to-Peer protocol (SMPP) is based on the exchange of request and response protocol data units (PDUs) between the Connection Manager acting as an external short message entity (ESME) and the short message service center (SMS-C). The Connection Manager supports all PDU operations that are defined in the SMPP Specification Version 3.4.

To configure a short message peer-to-peer message service, you must bring up the Gatekeeper interface and add a resource and selected Mobile Network Connection.

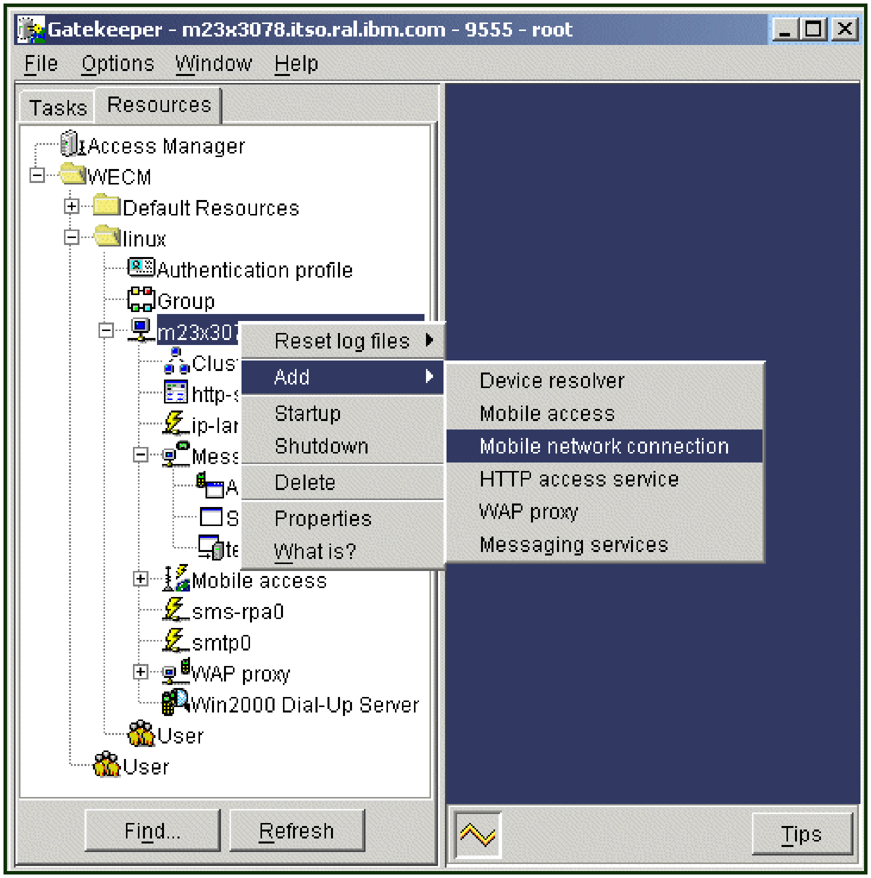
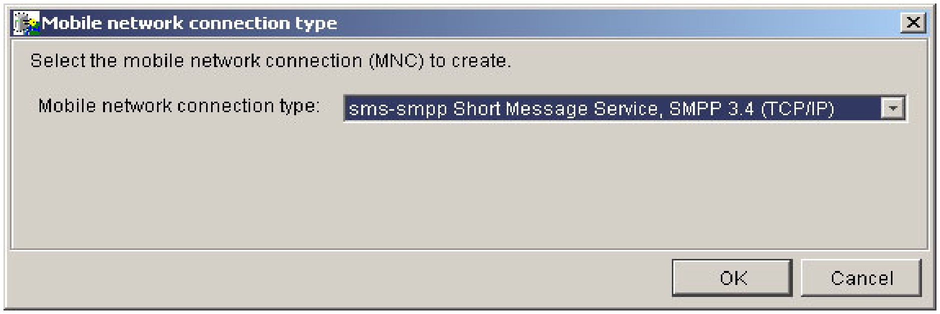


Figure 18-18 Add Mobile Network Connection (MNC)

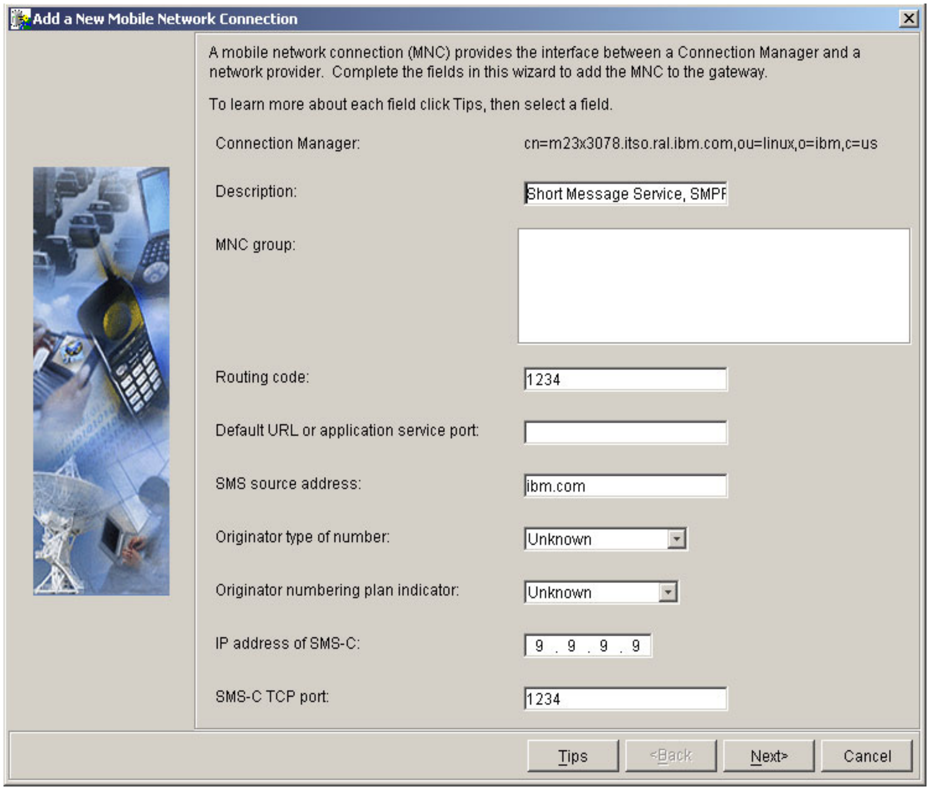
Next, select the **Mobile network connection type**.


Mobile network connection type

Select the mobile network connection (MNC) to create.

Mobile network connection type: sms-smpp Short Message Service, SMPP 3.4 (TCP/IP)

Figure 18-19 Mobile network connection type


Add a New Mobile Network Connection

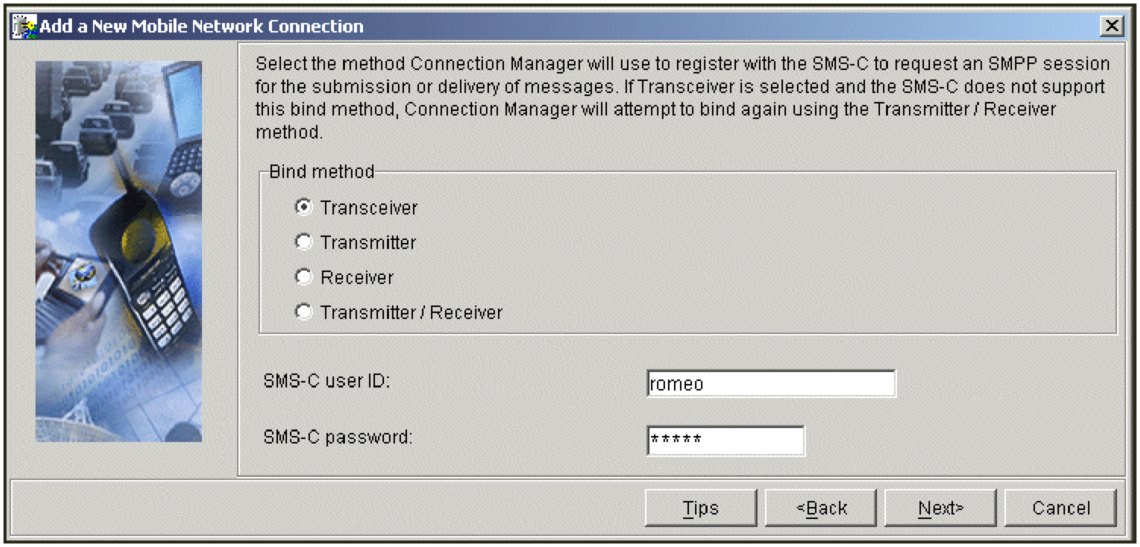
A mobile network connection (MNC) provides the interface between a Connection Manager and a network provider. Complete the fields in this wizard to add the MNC to the gateway.

To learn more about each field click Tips, then select a field.

Connection Manager:	cn=m23x3078.itso.ral.ibm.com,ou=linux,o=ibm,c=us
Description:	Short Message Service, SMPF
MNC group:	
Routing code:	1234
Default URL or application service port:	
SMS source address:	ibm.com
Originator type of number:	Unknown
Originator numbering plan indicator:	Unknown
IP address of SMS-C:	9 . 9 . 9 . 9
SMS-C TCP port:	1234

Figure 18-20   Adding a Mobile Network Connection (MNC)

Next, add the MNC to the gateway by providing an interface between the Connection Manager and the network. This is done by completing the MNC screen wizard.



Add a New Mobile Network Connection

Select the method Connection Manager will use to register with the SMS-C to request an SMPP session for the submission or delivery of messages. If Transceiver is selected and the SMS-C does not support this bind method, Connection Manager will attempt to bind again using the Transmitter / Receiver method.

Bind method

● Transceiver
○ Transmitter
○ Receiver
○ Transmitter / Receiver

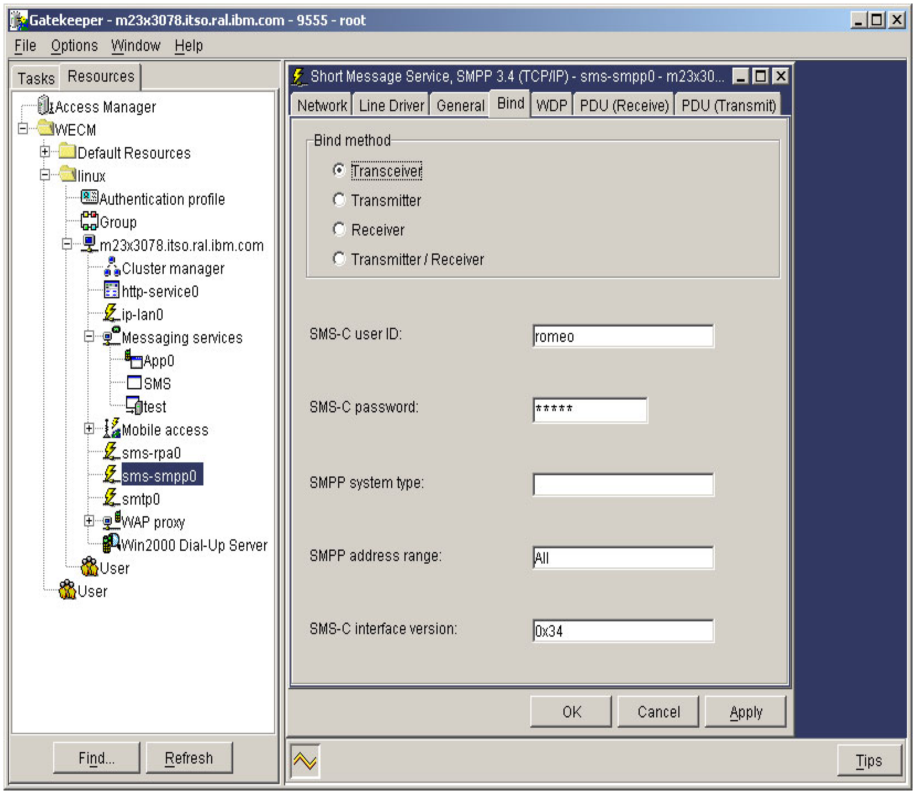
SMS-C user ID: romeo

SMS-C password: *****

Figure 18-21 Select bind method

The Connection Manager can attempt to register or bind with an SMS-C using one of the following methods:

- **Transceiver** - The Connection Manager sends messages to the SMS-C and receives messages from the SMS-C over a single SMPP session. If this method is selected and the SMS-C does not support it, the Connection Manager will attempt to bind again using the transmitter/receiver method.
- **Transmitter** - The Connection Manager sends short messages to the SMS-C and receives the corresponding SMPP responses from the SMS-C.
- **Receiver** - The Connection Manager receives short messages from the SMS-C and returns the corresponding SMPP message response to the SMS-C.
- **Transmitter/Receiver** - Uses two sessions to both send and receive short messages between the Connection Manager and the SMS-C.



Bind method

● Transceiver
○ Transmitter
○ Receiver
○ Transmitter / Receiver

SMS-C user ID: romeo

SMS-C password: *****

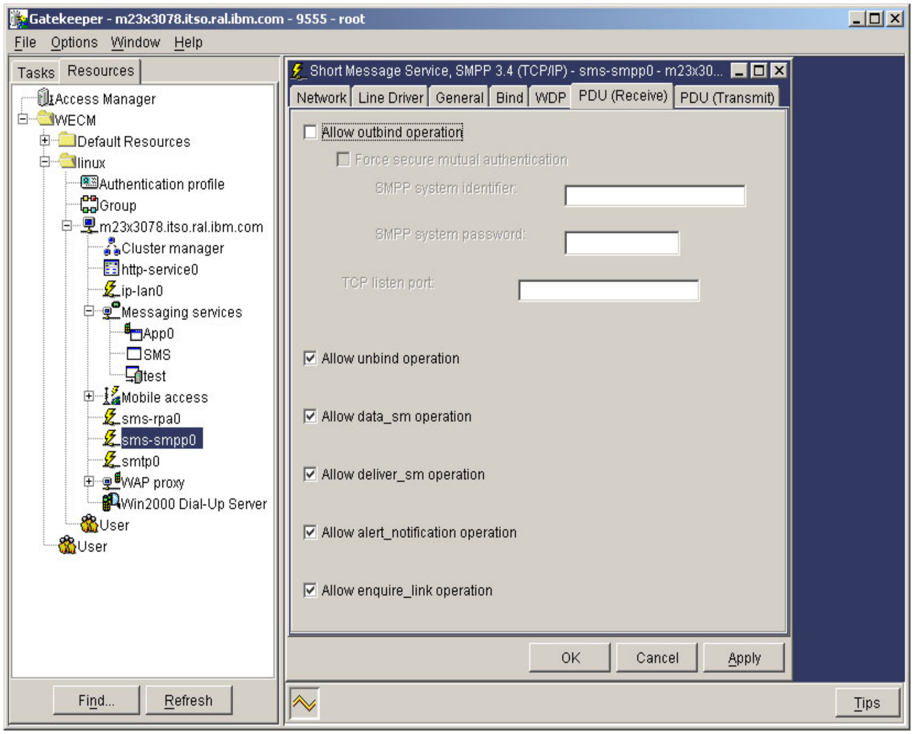
SMPP system type:

SMPP address range: All

SMS-C interface version: 0x34

Figure 18-22 Bind configuration

To view or change the bind method, edit the properties of the SMPP MNC. Click the **Bind** tab, then select the bind method to use for between this MNC and the SMS-C.



☐ Allow outbind operation

☐ Force secure mutual authentication

SMPP system identifier:

SMPP system password:

TCP listen port:

☑ Allow unbind operation

☑ Allow data_sm operation

☑ Allow deliver_sm operation

☑ Allow alert_notification operation

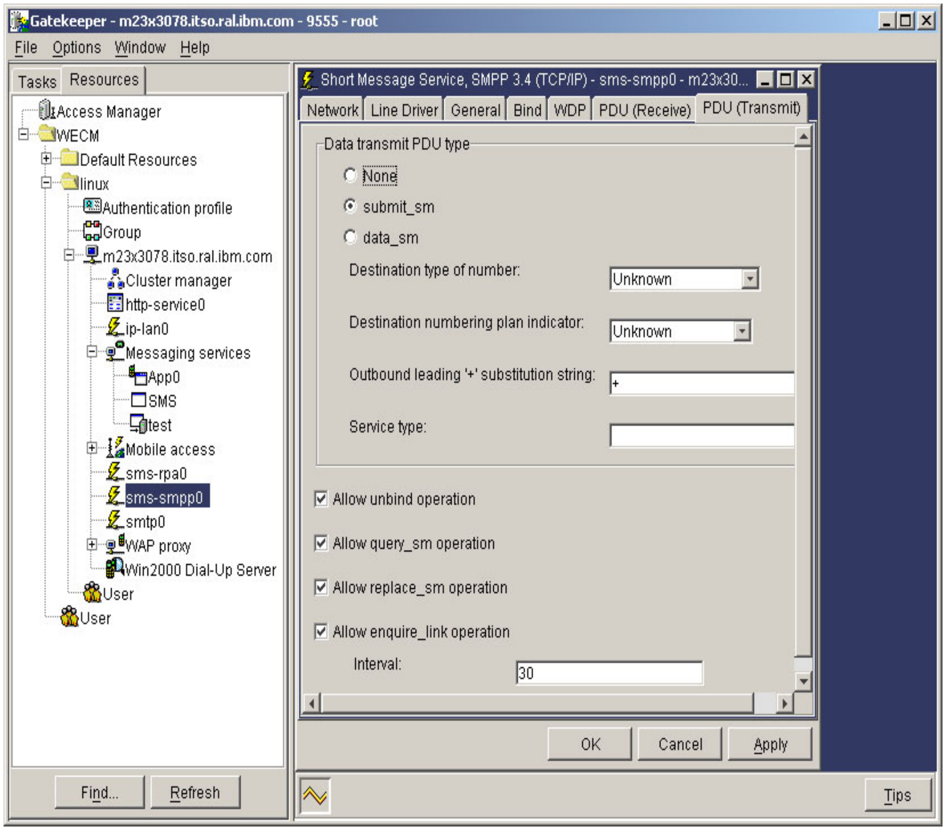
☑ Allow enquire_link operation

Figure 18-23 PDU (receive) configuration

To view or change the PDU operations when the Connection Manager is configured as a receiver, edit the properties of the SMPP MNC. Click the **PDU (Receive)** tab, then select the operations.

- outbind - Specifics whether the SMS-C is allowed to signal the Connection Manager to originate a bind_receiver request in which the Connection Manager can receive short messages from the SMS-C and returns the corresponding SMPP message responses to the SMS-C. This PDU operation can be configured to require that the connection is mutually authenticated.
- unbind - Specifies whether the Connection Manager or SMS-C can send a logoff request to close the current SMPP session
- data_sm - Specifies whether the Connection Manager is allowed to request that the SMS-C transfer message to an SMS client. The SMS-C may also use this operation to transfer a mobile-originated message to the Connection Manager.
- deliver_sm - Specifies whether the SMS-C routes short messages to the Connection Manager for delivery

- alert_notification - Specifies whether a message is sent by the SMS-C to the Connection Manager when the SMS-C has detected that a particular SMS client has become available, and a delivery pending flag had been set for that subscriber from a previous data_sm operation.
- enquire_link - Specific whether a message can be sent by either the Connection Manager or the SMS-C that is used to provide a confidence-check of the communication path between the Connection Manager and the SMS-C. On receipt of this request the receiving party sends a response that verifies that the application level connection between them is functioning.
- In addition, the data_sm and deliver_sm operations can be used to transfer the following types of special messages to the Connection Manager:
  - **SMS-C delivery receipt** - On detecting the final state of a registered message stored in the SMS-C, the SMS-C generates a receipt message addressed to the originator of the message.
  - **Short Message Entity (SME) delivery acknowledgement** - An indication from the recipient SME that the user has read the short message.
  - **SME manual/user acknowledgment** - An application-generated reply message sent in response to an application request message.
  - **Intermediate notification** - Provides an intermediate status of a message delivery attempt



Data transmit PDU type

○ None
● submit_sm
○ data_sm

Destination type of number: Unknown

Destination numbering plan indicator: Unknown

Outbound leading '+' substitution string: +

Service type:

☑ Allow unbind operation

☑ Allow query_sm operation

☑ Allow replace_sm operation

☑ Allow enquire_link operation

Interval: 30

Figure 18-24 PDU (transmit) configuration

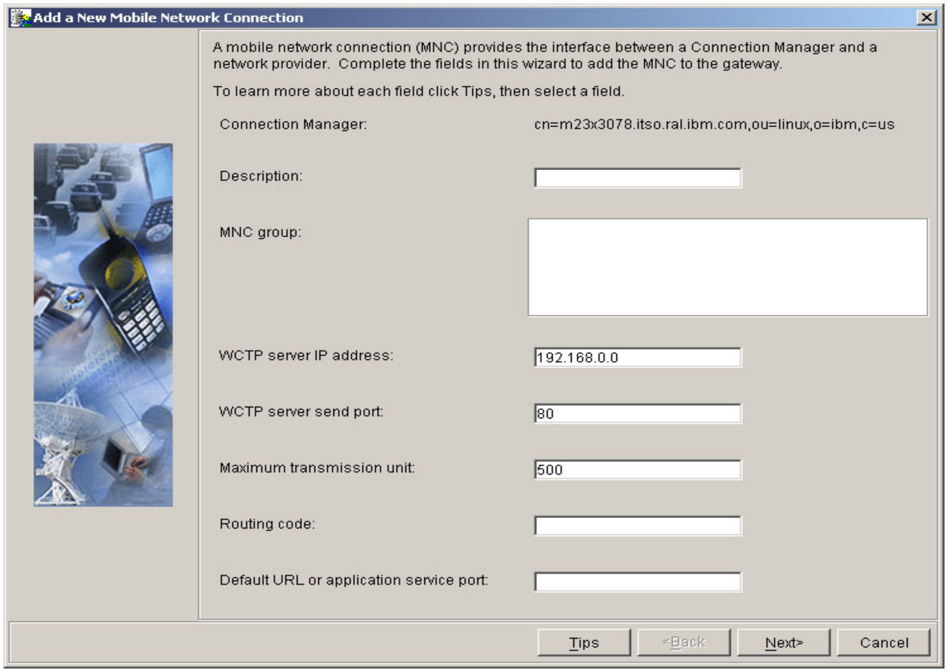
To view or change the PDU operations when the Connection Manager is configured as a transmitter, edit the properties of the SMPP MNC, click the **PDU (Transmit)** tab, then select the operations. When the Connection Manager is configured as a transmitter, it can be configured to allow these PDU operations:

- submit_sm - Allows the Connection Manager to submit a short message to the SMS-C for onward transmission to a specified short message entity (SME). This operation does not support the transaction message mode.
- data_sm - Specifics whether the Connection Manager is allowed to request that the SMS-C transfer a message to an SMS client. The SMS-C may also use this operation to transfer a mobile-originated message to the Connection Manager.
- unbind - Specifies whether the Connection Manager or SMS-C can send a logoff request to close the current SMPP session

- query_sm - Specifies that the Connection Manager is allowed to query the status of a previously submitted short message
- replace_sm - Specifies that the Connection Manager replaces a previously submitted short message that is still pending delivery
- enquire_link - Specifies wheather a message can be sent by either the Connection Manger or the SMS-C that is used to provide a confidence-check of the communication path between the Connection Manager and the SMS-C. On receipt of this request the receiving party sends a response that verifies that the application level connection between them is functioning.

Configuring a WCTP MNC

Wireless Communication Transfer Protocol (WCTP) is a wireless device messaging protocol that delivers wireless messages, both one and two-way to appropriate receiving devices such as pagers and cellular phones. When you configure a WCTP MNC, you provide the following as shown in Figure 18-25.



Add a New Mobile Network Connection

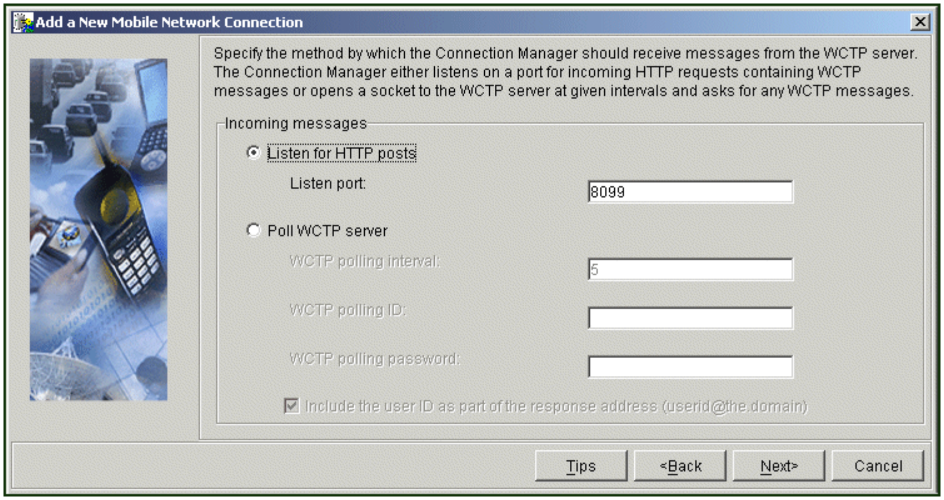
A mobile network connection (MNC) provides the interface between a Connection Manager and a network provider. Complete the fields in this wizard to add the MNC to the gateway.

To learn more about each field click Tips, then select a field.

Connection Manager:	cn=m23x3078.itso.ral.ibm.com,ou=linux,o=ibm,c=us
Description:	
MNC group:	
WCTP server IP address:	192.168.0.0
WCTP server send port:	80
Maximum transmission unit:	500
Routing code:	
Default URL or application service port:	

Figure 18-25 WCTP MNC configuration

- The WCTP server IP address and send port



Specify the method by which the Connection Manager should receive messages from the WCTP server. The Connection Manager either listens on a port for incoming HTTP requests containing WCTP messages or opens a socket to the WCTP server at given intervals and asks for any WCTP messages.

Incoming messages

● Listen for HTTP posts

Listen port: 8099

○ Poll WCTP server

WCTP polling interval: 5

WCTP polling ID:

WCTP polling password:

☑ Include the user ID as part of the response address (userid@the.domain)

Figure 18-26 Method to receive messages

- The method the Connection Manager should use to receive messages from the WCTP server. You have the option to choose from:
  - **Listen for HTTP post** - Requires a port number on which the Connection Manager listens for incoming HTTP request containing WCTP status and incoming messages. Choosing the method also requires that you identify a port number.
  - **Poll the WCTP server** - Requires a polling ID, password, and polling interval at which the Connection Manager polls the WCTP server for status and incoming messages. Choosing the method also requires that you identify a polling ID, polling password, and polling interval.
- Whether a user ID must be used along with the domain (for example, `user@domain`) as part of the response address in the send-responses-to field. Some carriers require the use of only the domain in response addresses, other carriers require that originator user IDs be appended to the domain.

18.4 Push communication

Connection Manager Messaging Services allows applications to send a push message to different types of client devices. Because WAP's PAP is general enough to also support other messaging technologies, IBM has added more delivery channels to use with push messages. Currently, messaging services supports the following delivery channels:

- WAP (using all WAP defined WDP bearers)

- SMTP
- GSM-SMS
- SNPP
- Mobitex SMS
- DataTAC
- RPA Wireless network
- WCTP

A WAP push operation starts when a Push Initiator (PI) sends a message to a Push Proxy Gateway (PPG) using PAP (Push Access Protocol). PAP contains control information, content, and optionally client capability information. PPG transfers the message to the client using the appropriate network.

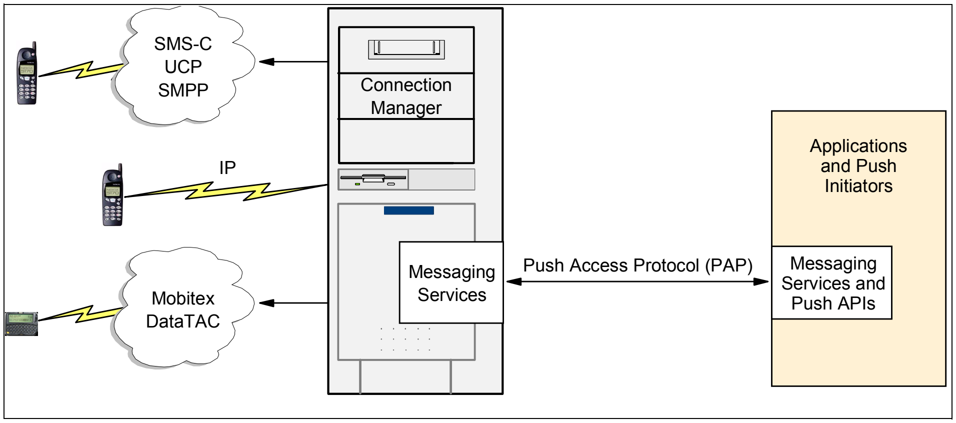

SMS-C
UCP
SMPP
Connection Manager
IP
Mobitex
DataTAC
Messaging Services
Push Access Protocol (PAP)
Applications and Push Initiators
Messaging Services and Push APIs

Figure 18-27 WAP push communication

In our case, Connection Manager Messaging Services is PPG and PAP messages are created using push APIs; see Figure 18-27.

The WAP push messages can be transferred to the client using all the available bearer networks that Connection Manager and the client support.

Note: To support WAP push, both messaging services and WAP proxy must be installed and configured properly.

18.5 Messaging toolkit examples

In following examples, we are using the Connection Manager Messaging toolkit to create few simple stand-alone Java applications to:

- Send push messages
- Query messages delivery status

- Receive mobile-originated messages

Examples are quite general, so they can be used also in other contexts, for example, in servlets and portlets.

You can download the Messaging toolkit from:

```
http://www-1.ibm.com/support/search.wss?q=toolkit&rs=804&tc=SSZQDW&dc=D
400
```

WAP push messages

In this section we the describe the WAP push messages.

WAP SI (Service Indication) push message

WAP SI message is used to inform and alert users without a client request. Optionally, a URI for the service to be loaded can be provided.

Let us create a simple Java stand-alone application, which sends a SI message to the WAP device. In our example, IPv4 address is used, so you can test your applications using your favorite push-enabled WAP emulator, for example, Nokia Mobile Internet Toolkit (NMIT). NMIT can be downloaded from `http://forum.nokia.com`. For examples on how to use SMS as a delivery channel, see “GSM SMS message” on page 403.

Example 18-1 SIPush.java

```
import com.ibm.wireless.push.*;

public class SIPush
{

   public static void main(String[] args)
   {

      try
      {
         String url="http://wap.company.com/mail";
         String msg="You have 4 new messages";
         String action="signal-high";
         WAPSIContent msgContent = new WAPSIContent(url, msg);

         PushMessage message = new PushMessage();
         message.setContent(msgContent);
         System.out.println(message);

         // Required for Nokia Mobile Internet Toolkit
         message.setXWapApplicationId("0");
         IPv4PushAddress address = new IPv4PushAddress(null);
         address.setAddress("emulator.company.com");

         Pusher pusher = new Pusher("http://wecm.company.com:13131", "");
         PushResponse response = pusher.push(message, address);
         System.out.println(response);

      }
      catch (PushException e)
      {
         e.printStackTrace();
      }

   }
}
```

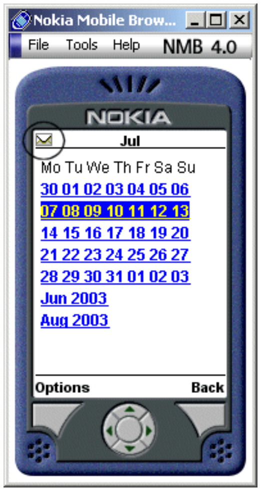
Example 18-1 creates a SI message, which contains informative messages “`You have 4 new messages`”, and an URI to company.com’s mail system `http://wap.company.com/mail`. Message content is added to a message object. Then an address object is created and IP address or host name of emulator is added to it.

A pusher object is created using address of Connection Manager gateway and messaging gateway port. Finally, message is sent using pusher object and response similar in Example 18-2 is printed out to System.out.

Example 18-2 Push response output

```
Push Response
=============
Push ID: aEEhfPld+iUMsQCKLwDM3Q==
Product name: IBM WebSphere Everyplace Connection Manager V5.0
Sender name:
Sender address: http://m23x3078:13131
Reply time: Fri Jul 18 10:05:59 EDT 2003
Description: Accepted for Processing
Status short description:
Status detailed description:
```

Now, let us see what happends when push-message arrives to the client device.

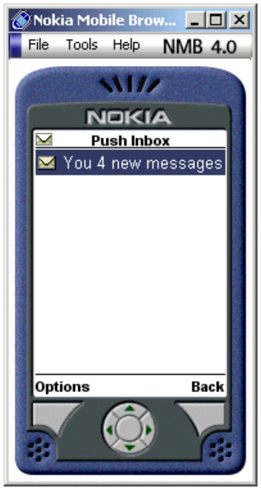


Jul

Mo Tu We Th Fr Sa Su
30 01 02 03 04 05 06
07 08 09 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31 01 02 03
Jun 2003
Aug 2003

Figure 18-28 Push message arrives to device

User is browsing calendar when one notices that a push message has arrived. (Figure 18-28).

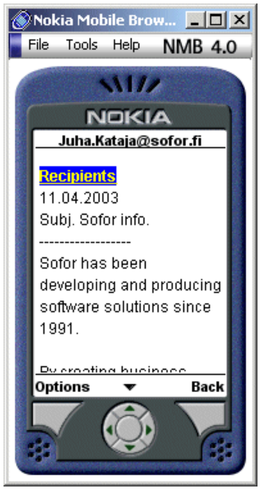


Push Inbox

You 4 new messages

Figure 18-29 Push inbox opened

The user opens the push inbox and sees our push message (Figure 18-29).



Juha.Kataja@sofor.fi

Recipients
11.04.2003
Subj. Sofor info.

Sofor has been developing and producing software solutions since 1991.

By creating business

Figure 18-30 Mail system opened

The user opens the push message and browser loads the user's mail (Figure 18-30).

If the push message did not arrive to your emulator, check from `Messaging Service properties` that default port on WAP client `to accept WAP push messages:` is the same as your emulator is listening to.

For more information about configuring messaging services, see 18.3, "Messaging services configuration" on page 376.

WAP SL (service loading) push message

The WAP SL message is used to inform and alert the user by sending a service URI to the client, which the client browser then loads. Messages can be intrusive or non-intrusive.

Let us create a simple Java stand-alone application, which sends a SL message to the WAP device. In our example, IPv4 address is used, so you can test your applications using your favorite push-enabled WAP emulator, for example, Nokia Mobile Internet Toolkit (NMIT). NMIT can be downloaded from:
http://forum.nokia.com
For examples on how to use SMS as a delivery channel, see "GSM SMS message" on page 403.

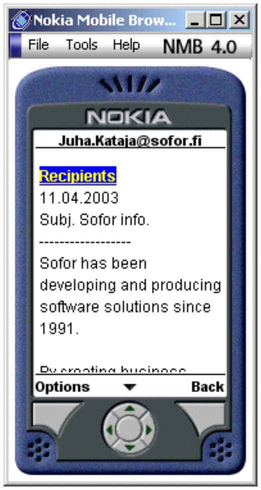
Example 18-3 SLPush.java

```
import com.ibm.wireless.push.*;

public class SLPush
```

```
{

   public static void main(String[] args)
   {

      try
      {
         String url="http://wap.company.com/mail";
         String action="execute-high";
         WAPSLContent msgContent = new WAPSLContent(url);

         PushMessage message = new PushMessage();
         message.setContent(msgContent);

         // Required for Nokia Mobile Internet Toolkit
         message.setXWapApplicationId("0");

         IPv4PushAddress address = new IPv4PushAddress(null);
         address.setAddress("emulator.company.com");

         Pusher pusher = new Pusher("http://wecm.company.com:13131", "");
         PushResponse response = pusher.push(message, address);
         System.out.println(response);

      }
      catch (PushException e)
      {
         e.printStackTrace();
      }

   }
}
```

Example 18-3 creates a SL message, which contains an URI to company.com's mail system `http://wap.company.com/mail`. Message content is added to a message object. Then an address object is created and IP address or host name of an emulator is added to it. A pusher object is created using the address of the Connection Manager gateway and messaging gateway port. Finally, the message is sent using the pusher object and response similar to Example 18-2, which is printed out to System.out.

Unlike when receiving the SI message, the client browser loads the mail page automatically (Figure 18-31).



Juha.Kataja@sofor.fi

Recipients
11.04.2003
Subj. Sofor info.

Sofor has been developing and producing software solutions since 1991.

By creating business

Figure 18-31 Mail system opened

GSM SMS message

The SMS message is used to inform and alert user using SMS, whenever the client device is reachable by the bearer network.

Let us create a simple Java stand-alone application, which sends a SMS message to the GSM device using the phone number.

> **Note:** The SMS service must be installed and configured before sending SMS messages.

Example 18-4 SMSPush.java

```
import com.ibm.wireless.push.*;

public class SMSPush
{

   public static void main(String[] args)
   {

      try
      {
         String msgContent = "Hello!";

         PushMessage message = new PushMessage();
         message.setContent(msgContent,"text/plain");
         PLMNPushAddress address = new PLMNPushAddress(null);
         address.setAddress("+1-234-456-789");

         Pusher pusher = new Pusher("http://wecm.company.com:13131", "");
         PushResponse response = pusher.push(message, address);
         System.out.println(response);

      }
      catch (PushException e)
      {
         e.printStackTrace();
      }

   }
}
```

Example 18-4 creates a SMS message, which contains simple message, `Hello!`. The message content is added to a message object using the appropriate content type, which for SMS message is `text/plain`. Then an address object is created and the client's phone number is added to it.

A pusher object is created using the address of the Connection Manager gateway and messaging gateway port. Finally, the message is sent using the pusher object, and the response is similar to Example 18-2 on page 399, which is printed out to System.out.

The SMS message appears in the client device as it is a SMS message sent by another user (Figure 18-32).

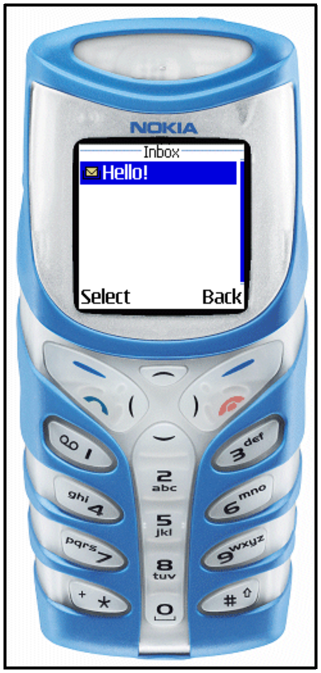


Figure 18-32 SMS inbox

SMTP message

The SMTP message is used to inform and alert the user using e-mail.

Let us create a simple Java stand-alone application, which sends a SMTP message to the user using an e-mail address.

> **Note:** SMTP messaging must be installed and configured before sending SMTP messages.

Example 18-5 SMTPPush.java

```
import com.ibm.wireless.push.*;

public class SMTPPush
{

   public static void main(String[] args)
   {

      try
      {
         String msgContent = "Hello!";

         PushMessage message = new PushMessage();
         message.setContent(msgContent,"text/plain");

         SMTPPushAddress address = new SMTPPushAddress(null);
         address.addToRecipient("someone@company.com");

         Pusher pusher = new Pusher("http://wecm.company.com:13131", "");
         PushResponse response = pusher.push(message, address);
         System.out.println(response);

      }
      catch (PushException e)
      {
         e.printStackTrace();
      }

   }
}
```

Example 18-5 creates an e-mail message, which contains simple message, “`Hello!`”. Message content is added to a message object using the appropriate content type, which in our example is `text/plain` but in SMTP message, any MIME type can be used. Then an address object is created and the users e-mail address is added to recipients (To -field).

A pusher object is created using the address of the Connection Manager gateway and messaging gateway port. Finally, message is sent using pusher object and response similar to Example 18-2 on page 399, which is printed out to System.out.

Querying message status

In any point after the message has been submitted, delivery status of the message can be queried using the queryStatus method:

```
StatusResponse response = pusher.queryStatus(<pushID>, null);
```

where *<pushID>* is a push ID of message to be queried. If you print the response object to System.out using:

```
System.out.println(response);
```

you will see output similar to Example 18-6.

Example 18-6 Delivery status query response

```
Status Response
===============
Push ID: eBWbOBxG31/RWYJWCD3tvw==
Product name: IBM WebSphere Everyplace Connection Manager V5.0

Status Result
-------------
Valid for addresses:
wappush=+1-234-456-789/type=PLMN@wecm.company.com
Message state: delivered
Event time: Fri Jul 18 12:06:04 EDT 2003
Status short description:
Status detailed description:
Description: OK
Quality-of-service:
   Priority: medium
   Delivery method: notspecified
   Network required: false
   Bearer required: false
```

Note: You control which messages are kept in the database, and therefore can be queried on the Database tab of messaging services' properties.

Receiving mobile-originated messages

The messaging services can forward mobile-oriented messages to a URL by HTTP post, or to a specific application service.

Receiving mobile-originated messages using messagingtoolkit

In our example, user sends a SMS messages to Connection Manager and Connection Manager forwards the message to our Java application using the HTTP post (Figure 18-33).

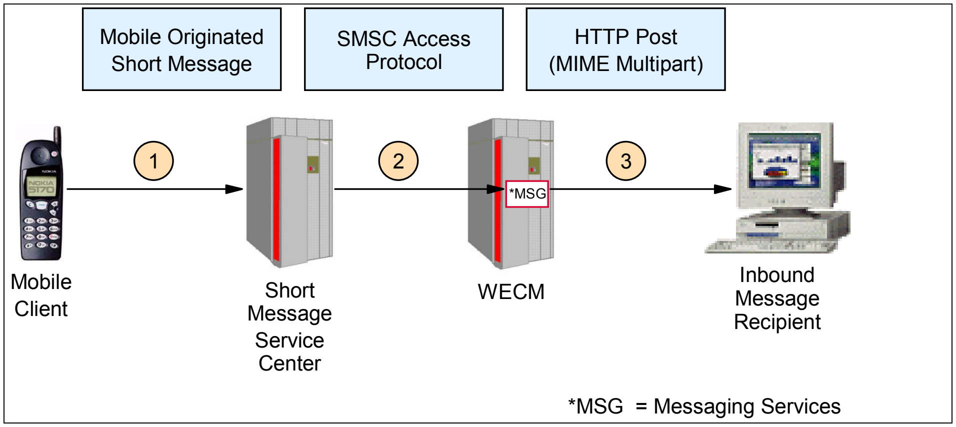
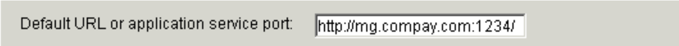

Mobile Originated Short Message
SMSC Access Protocol
HTTP Post (MIME Multipart)
1
2
3
*MSG
Mobile Client
Short Message Service Center
WECM
Inbound Message Recipient
*MSG = Messaging Services

Figure 18-33 Receiving mobile-originated messages

To enable HTTP message forwarding, check the configuration of the used SMS Mobile Network Connection (MNC). On the Network tab, check that the default URL for application port is set as in Figure 18-34.



Default URL or application service port: http://mg.compay.com:1234/

Figure 18-34 Default URL

Now, let us write a stand-alone Java application using messaging toolkit APIs, which wait for incoming message, print them out, and then stops the receiver.

Example 18-7 MSGReceive.java

```
import java.io.IOException;
import com.ibm.wireless.push.*;

public class MSGReceive implements MessageListener
{
   private static int portNumber = 1234;

   public static void main(String[] args)
   {

      MSGReceive dump  = new MSGReceive();
      MessageReceiver rx = null;

      try
      {
         rx=MessageReceiver.getInstance(portNumber);
      }
      catch (IOException e)
      {
         e.printStackTrace(); // handle exception
      }

      rx.addMessageListener(dump);
      System.out.println("Waiting for incoming message ...");

   }

   public void notifyMessage(ReceivedMessage msg)
   {

      System.out.println(msg.toString());
      try
      {
         MessageReceiver rx = MessageReceiver.getInstance(portNumber);
         rx.stopReceiver();
      }
      catch (Exception e)
      {
         e.printStackTrace(); // handle exception
      }

   }

}
```

In Example 18-7, in the `main` method, we first created a new instance of a `MSGReceive` class, which implements the `MessageReceiver` interface. Then we create a new instance of the `MessageReceiver` to listen to a specific port. Next, we register our `MessageListener` to listen for incoming messages using the `addMessageListener` method.

Whenever a message in coming to a listened port, `MessageReceiver` creates a new thread for a registered `MessageListener` and calls its `notifyMessage` method.

In `notifyMessage` method, we print out a received message, then get instance of `MessageReceiver`, and finally tell it to stop listening using the `stopReceiver` method.

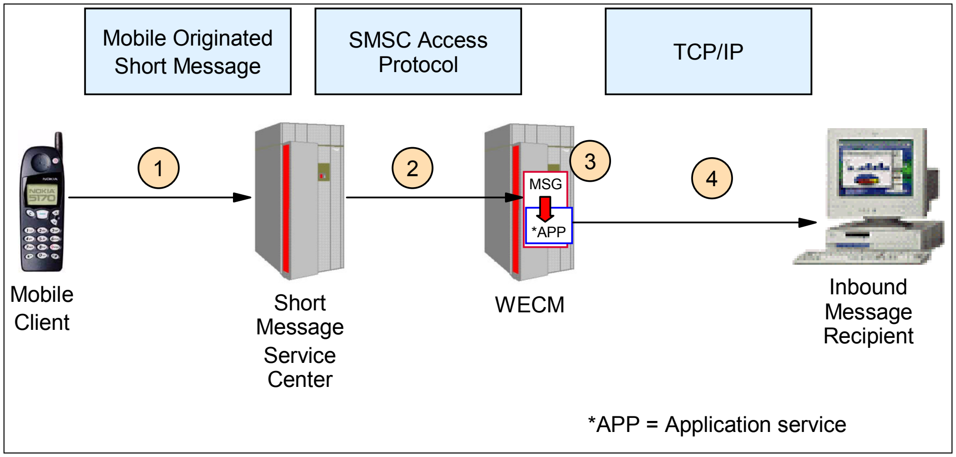
When you start the application and a message `Hello!` is received, output similar to Example 18-8 is printed out.

Example 18-8 MSGReceive output

```
Waiting for incoming message ...

Message
=======
Content:
-------
Hello!

Originator: +123-456-789
Destination: +1-23-345
Posted to: http://ms.company.com:1234
Product name: IBM WebSphere Everyplace Connection Manager V5.0
Network type: PLMN
MTA name: sms-smpp0
WCTP Settings
Transaction ID:
ReplyTo ID:
WCTP mesage type: ALPHANUMERIC
```

Receiving mobile-originated messages using Java sockets

In this example, the user sends a SMS messages to Connection Manager and Connection Manager forwards the message to a specific application service, which then adds the device information to message and forwards the message to our Java application (Figure 18-35).



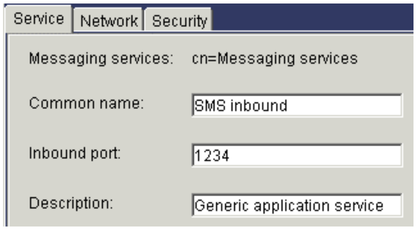
Mobile Originated Short Message
SMSC Access Protocol
TCP/IP
1
2
3
4
MSG
*APP
Mobile Client
Short Message Service Center
WECM
Inbound Message Recipient
*APP = Application service

Figure 18-35 Mobile-originated messages using Application service

To enable the application service message forwarding, you must configure an application service for SMS.

First, you need to create an application service that listens for incoming traffic and forwards it to our application. These are the steps:

1. Create the new application service using Gatekeeper.
2. On the Service tab, enter a common name for service and an inbound port number. This is the port where SMS MNC will forward all incoming messages (Figure 18-36).
3. On the Network tab, select **TCP** as the transfer protocol. Enter the hostname and the port number of the application server. Use the address of the computer where you will run the sample application. Our sample application will parse device identification headers, so be sure that **Include device identifier information in data stream** is checked (see Figure 18-37).
4. On the Security tab, you can select **None** for authentication type. Be sure to check at least the MNC that you are using to receive SMS messages from the Mobile Network Connection list (Figure 18-38).



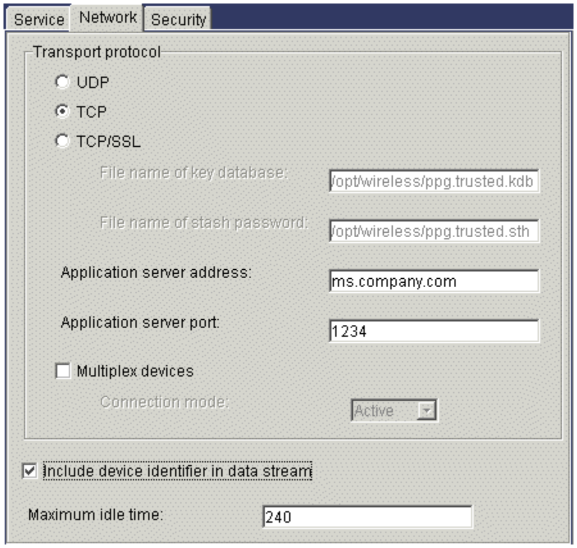
Messaging services: cn=Messaging services

Common name: SMS inbound

Inbound port: 1234

Description: Generic application service

Figure 18-36 Service settings



Transport protocol

○ UDP

● TCP

○ TCP/SSL

File name of key database: /opt/wireless/ppg.trusted.kdb

File name of stash password: /opt/wireless/ppg.trusted.sth

Application server address: ms.company.com

Application server port: 1234

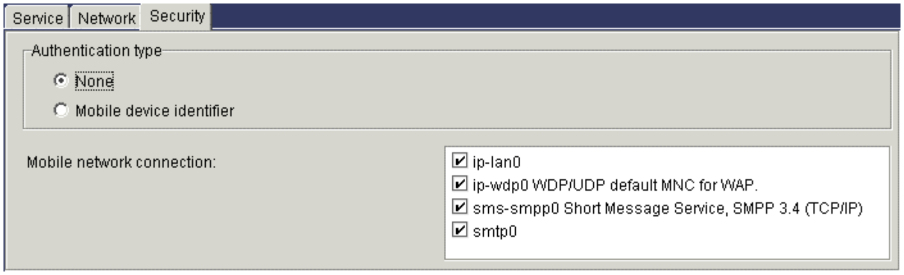
□ Multiplex devices

Connection mode: Active

☑ Include device identifier in data stream

Maximum idle time: 240

Figure 18-37 Network settings



Authentication type

● None

○ Mobile device identifier

Mobile network connection:

☑ ip-lan0

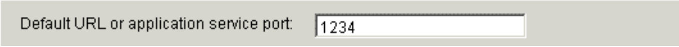
☑ ip-wdp0 WDP/UDP default MNC for WAP.

☑ sms-smpp0 Short Message Service, SMPP 3.4 (TCP/IP)

☑ smtp0

Figure 18-38 Security settings

Next, you need to check configuration of the used Mobile Network Connection (SMS MNC). On the Network tab, check that the application port number is the same that you used when you created an application service (Figure 18-39).



Default URL or application service port: 1234

Figure 18-39 Default application service port

Now, let us write a stand-alone Java application using the Java socket interface. The application waits for an incoming message, prints it out, and then exits.

Example 18-9 SocketReceiver.java

```
import java.io.*;
import java.net.*;

public class SocketReceiver implements Runnable
{
   private Socket socket=null;

   public static void main(String[] args)
   {
      try
      {
         ServerSocket server = new ServerSocket(1234);

         Socket socket = server.accept();
         new Thread(new SocketReceiver(socket)).start();

         server.close();
      }
      catch (Exception e)
      {
         e.printStackTrace();
      }

   }

   public SocketReceiver(Socket in)
   {
      this.socket=in;
   };

   public void run()
   {
      try
      {
         InputStream input = socket.getInputStream();
         int available = input.available();
         byte buf[]=new byte[available];
         input.read(buf);

         int packetLength=buf[1]+buf[0]*256;
         int infoLength=buf[2];
         int deviceType=buf[3];
         int numberLength=buf[4];
         int deviceType2=buf[5];

         String phoneNumber=new String(buf,6,numberLength-1);
         String message=new String(buf,2+infoLength, packetLength-infoLength);
```

```
            System.out.println("Message from "+phoneNumber+": "+message);

            socket.close();
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }

    }
}
```

In Example 18-9 in the `main` method, we first created a new instance of a `ServerSocket` class. Then we called the `accept` method, which waits for a connection, and whenever Connection Manager connects to given port, it receives a new instance of the `Socket` object. Then we constructed the new `SocketReceiver` by giving received `socket` to it as a parameter, and started a new thread for it.

In the `run` method we read the incoming message to a buffer, and then we parsed the device identifier header to receive the sender phone number, and finally printed out the message similar to the following:

```
Message from +123-456-789: Hello!
```

19

WAP gateway

In this chapter we introduce WAP communication and describe the WAP gateway (or proxy) installation and configuration:

- WAP gateway installation
- Additional WAP service installation
- WAP gateway message book utility
- IP Wireless Datagram Protocol configuration (ip-wdp)
- WML transcoding and fragmentation

> **Note:** For information about WAP security, see Chapter 20, “WAP security” on page 437.

19.1 WAP communications

The main function of the WAP proxy (gateway) is to perform a protocol conversion between wireless protocols and HTTP protocols. This conversion enables a communication between WAP clients and HTTP servers.

The WAP proxy listens for data from WAP clients and translates the Wireless Session Protocol (WSP) into Hyper Text Transfer Protocol (HTTP) requests and forwards them to a HTTP proxy. The response headers are converted from HTTP headers to WSP headers, and content is encoded from Wireless Markup Language (WML) and WMLScript to binary WML format (wbxml) and sends it back to the client.

When content is converted from WML to binary WML, it is also compressed. Every WML tag is replaced by tokens, which reduces the size of the binary output. Some WAP gateways such as Connection Manager WAP proxy can also recognize commonly used strings and replace those strings with tokens too. For example, if there are 10 links in an WML document that all begin with `http://www.company.com/wps/portal`, the string appears in the binary document only once, others are replaced by tokens. This gives approximately 300 bytes more of space to use with the more important content.

WAP proxy supports all WAP defined WDP bearers (see Figure 19-1).

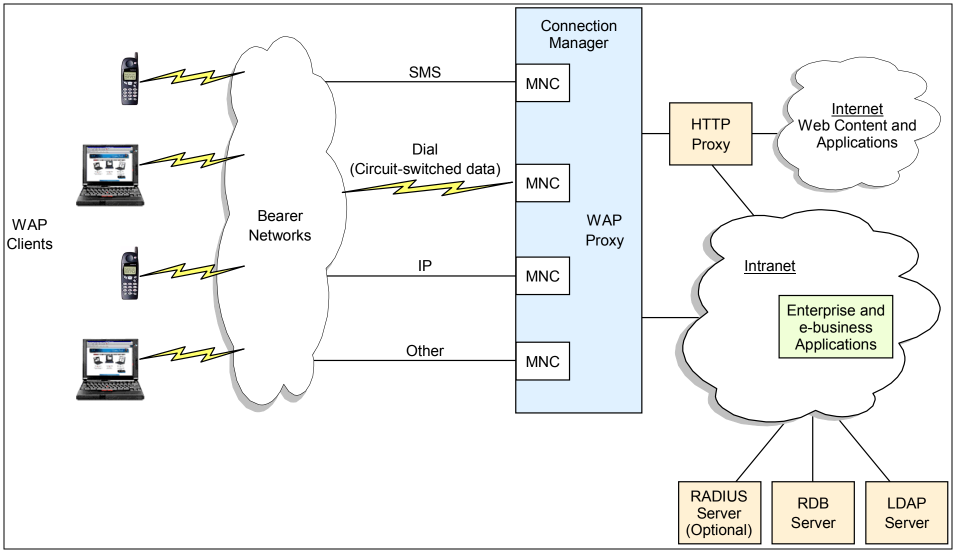


Figure 19-1 WAP communication

There are four types of WAP service protocols:

- **Connectionless**

  Asynchronous and not acknowledged unless there is a persistent session between WAP client and the WAP proxy. Default port is 9200

- **Connection-oriented**

  Establishes a persistent session between the WAP clients and the WAP proxy before any messages are transmitted. Messages are typically acknowledged. The default port is 9201.

- **Secure connectionless**

  Asynchronous and not acknowledged, unless there is a persistent session between WAP client and the WAP proxy. All data between the WAP clients and the WAP proxy is encrypted using Wireless Transport Layer Security (WTLS) protocol. The default port is 9202.

- **Secure connection oriented**

  Establishes a persistent session between the WAP clients and the WAP proxy before any messages are transmitted. Messages are typically acknowledged. All data between the WAP clients and the WAP proxy is encrypted using WTLS protocol. The default port is 9203.

WAP client Web request headers

The WAP proxy adds and modifies HTTP headers of the WAP request to give more information about the WAP device to back-end servers. Some of these headers are required by WebSEAL-Lite. For example:

- Through

  `<protocol>/1.2 <gw_hostname> (Connection Manager/5.0.0.0)`

  where *<protocol>* is WSP or WSPS if WTLS connection is used. *<gw_hostname>* is the hostname of Connection Manager without the DNS suffix.

  Example:

  ```
  Via: WSP/1.2 m23x3078 (Connection Manager/5.0.0.0)
  ```

- X-NetWork-Info

  `<bearer_name>`,*<wap_device_address>*,security=*<seclevel>*

  where <bearer_name> is name of used MNC. <wap_device_address> is the address of the WAP device, can be for example an IP address or a phone number depending on used bearer. <seclevel> specifies how secure the WTLS connection is. 0 = no WTLS, 1 = weak, 2 = medium, 3 = strong.

  Example:

  ```
  X-NetWork-Info: ip-wdp0,10.1.2.3,security=0
  ```

- X-IBM-PVC-User

  <userid>@<realm>

  where *<userid>* is user ID and *<realm>* is DN of the user's Organizational Unit.

  Example:

  ```
  X-IBM-PVC-User: max@ou=linux,o=ibm,c=us
  ```

  Note: X-IBM-PVC-User header is added to request only if the user identification is set to `User authentication` or `External account lookup`.

- X-IBM-PVC-Client-Id

  <RADIUS_attribute><clientid>

  <RADIUS_attribute> is 2-digit uppercase hexadecimal presentation of RADIUS attribute code which identifies the type of *<clientid>*.*<clientid>* is the ID of the used WAP device, which usually is the IP address or phone number depending on the used bearer. Also, if a device resolver is configured, *<clientid>* can be any value returned by RADIUS. The most often used RADIUS attributes are listed in Table 19-1.

Table 19-1 RADIUS attribute codes

RADIUS attribute	Description
0x01	User ID of the WAP phone
0x08	Dotted-Decimal IP address of the WAP client
0x0F	Phone number

  Note: When using some WebSphere Everyplace Suite components, WebSEAL-Lite expects to find the IP address of the WAP phone from X-IBM-PVC-Client-Id.

  Note: If the X-PVC-Client-Id header is configured to be encrypted, only the *<clientid>* part of header is encrypted, not the RADIUS attribute.

  Example:

  ```
  X-IBM-PVC-Client-Id: 08192.168.0.123
  ```

- X-IBM-PVC-Network-Type

  `wireless`

This header always contains the value wireless.

Example:

```
X-IBM-PVC-Network-Type: wireless
```

- User-Agent

  `unknown WAP Device`

  > **Note:** User-Agent is added to the request only if the WAP device does not supply its own User-Agent header.

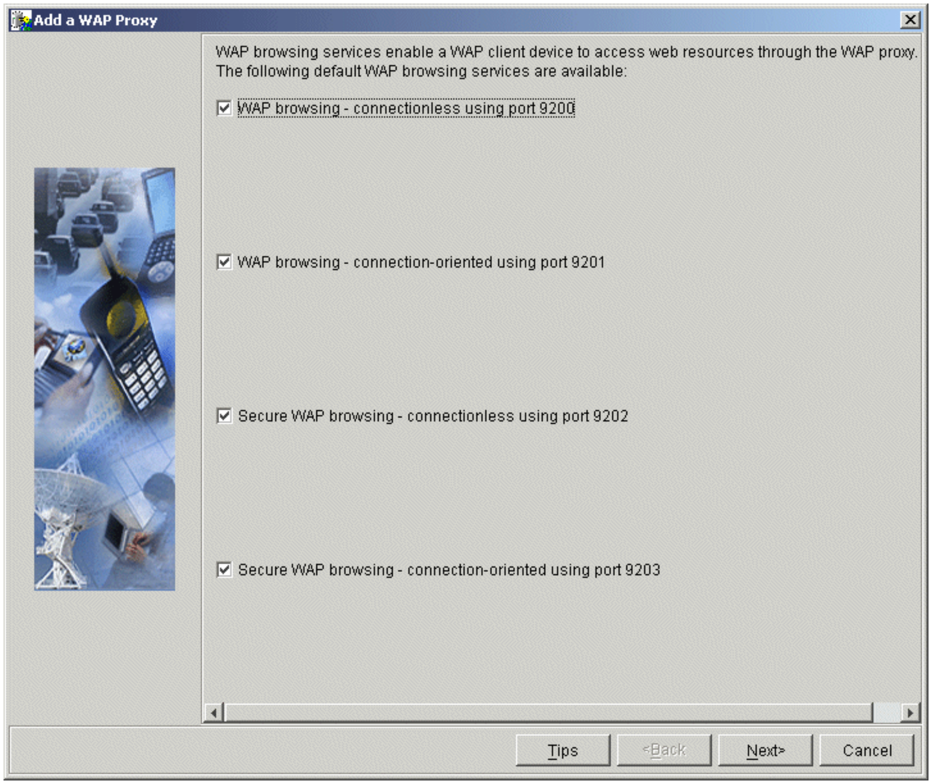
  Example:

  ```
  User-Agent: unknown WAP Device
  ```

19.2 WAP proxy installation

You can install WAP proxy at the same time you install your Connection Manager, or afterwards using Gatekeeper.

To start installation from Gatekeeper, select the **Connection Manager** on which you would like WAP proxy to be installed. Then right-click and select **Add -> WAP proxy**.



WAP browsing services enable a WAP client device to access web resources through the WAP proxy. The following default WAP browsing services are available:

☑ WAP browsing - connectionless using port 9200

☑ WAP browsing - connection-oriented using port 9201

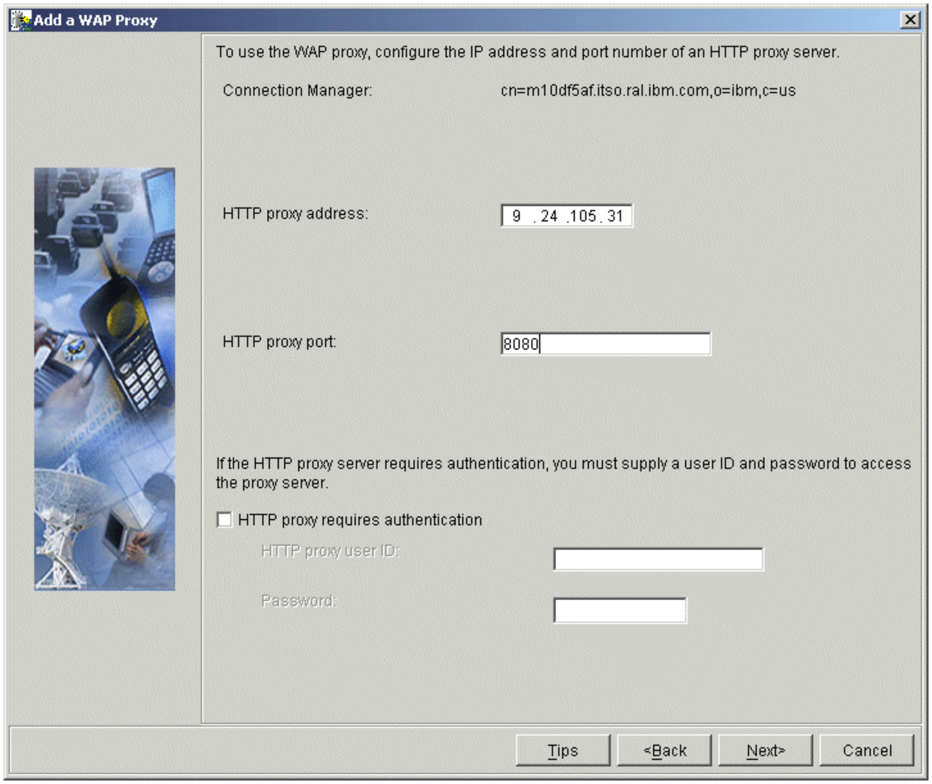
☑ Secure WAP browsing - connectionless using port 9202

☑ Secure WAP browsing - connection-oriented using port 9203

Figure 19-2   WAP service selection

Select which WAP services you are planning to use. See Figure 19-2. For more information about WAP service protocols, see “There are four types of WAP service protocols:” on page 416.

If you need to add more WAP services, see 19.2.1, “Additional WAP service installation” on page 423.



To use the WAP proxy, configure the IP address and port number of an HTTP proxy server.

Connection Manager: cn=m10df5af.itso.ral.ibm.com,o=ibm,c=us

HTTP proxy address: 9 . 24 .105 . 31

HTTP proxy port: 8080

If the HTTP proxy server requires authentication, you must supply a user ID and password to access the proxy server.

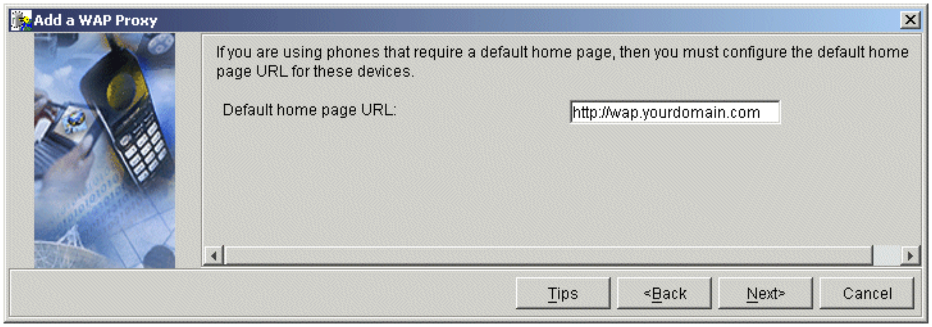
□ HTTP proxy requires authentication

HTTP proxy user ID:

Password:

Figure 19-3 HTTP proxy setup

Enter the IP address and port number of the HTTP proxy server that WAP proxy is connecting to. If the proxy server needs authentication, provide a valid user ID and password. See Figure 19-3.

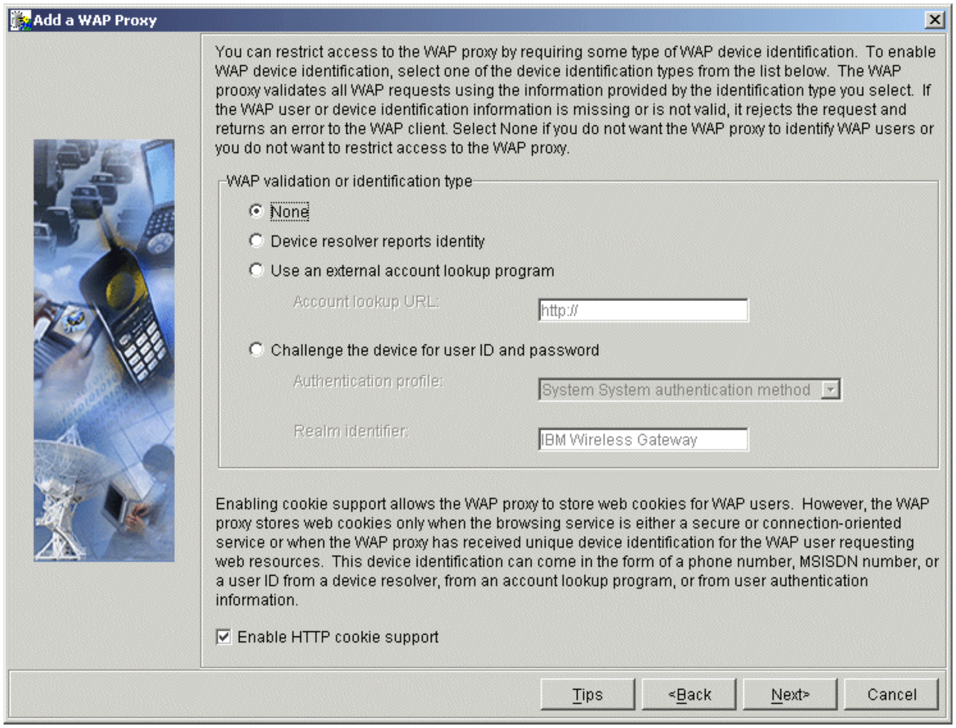


If you are using phones that require a default home page, then you must configure the default home page URL for these devices.

Default home page URL: http://wap.yourdomain.com

Figure 19-4 Default home page URL

Some WAP clients, for example, Phone.com microbrowsers, require a default home page. If those clients are used, insert the URL for the default home page. See Figure 19-4.



You can restrict access to the WAP proxy by requiring some type of WAP device identification. To enable WAP device identification, select one of the device identification types from the list below. The WAP proxy validates all WAP requests using the information provided by the identification type you select. If the WAP user or device identification information is missing or is not valid, it rejects the request and returns an error to the WAP client. Select None if you do not want the WAP proxy to identify WAP users or you do not want to restrict access to the WAP proxy.

WAP validation or identification type

● None

○ Device resolver reports identity

○ Use an external account lookup program

Account lookup URL: http://

○ Challenge the device for user ID and password

Authentication profile: System System authentication method

Realm identifier: IBM Wireless Gateway

Enabling cookie support allows the WAP proxy to store web cookies for WAP users. However, the WAP proxy stores web cookies only when the browsing service is either a secure or connection-oriented service or when the WAP proxy has received unique device identification for the WAP user requesting web resources. This device identification can come in the form of a phone number, MSISDN number, or a user ID from a device resolver, from an account lookup program, or from user authentication information.

☑ Enable HTTP cookie support

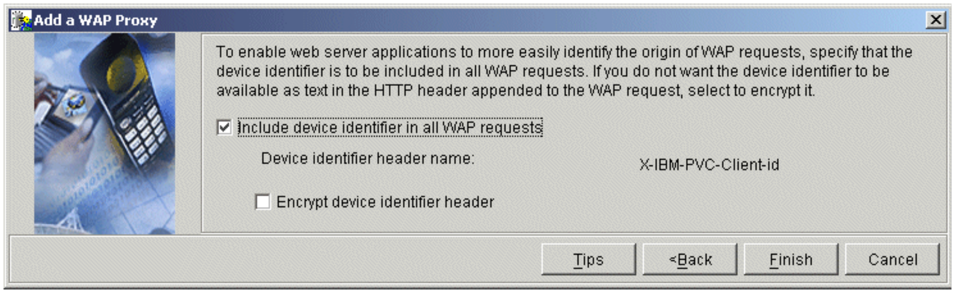
Figure 19-5 WAP identification and validation

If you need to restrict the access of your WAP clients, enable the WAP identification or validation, otherwise you can use the default values. Also, if billing and accounting statistics need to be logged for each device or user, enable WAP identification. For more information about WAP identification or validation, see Chapter 20, “WAP security” on page 437.

Not all WAP devices are capable of using HTTP cookies. WAP proxy can cache cookies for WAP clients if a secure or connection-oriented service is used. Also, if WAP requests contain some type of unique device identification (for example, a user ID) cookies can be cached by WAP proxy. HTTP cookies are used by applications and Web servers to track session information. If you need cookies, enable HTTP cookie support.

Note: If a secure or connection-oriented service is used, and WAP validation and identification is not used, only session cookies are stored by the WAP proxy.

See Figure 19-6.



Add a WAP Proxy

To enable web server applications to more easily identify the origin of WAP requests, specify that the device identifier is to be included in all WAP requests. If you do not want the device identifier to be available as text in the HTTP header appended to the WAP request, select to encrypt it.

☑ Include device identifier in all WAP requests

Device identifier header name: X-IBM-PVC-Client-id

☐ Encrypt device identifier header

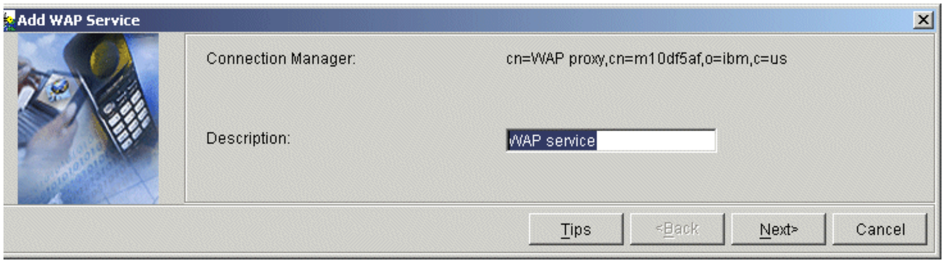
Figure 19-6 Device identifier

If you want Web servers and applications to be able to track more easily the origin of the WAP requests, enable the option to include the device identifier in all WAP requests. You can also select whether the identifier header is encrypted or not. See Figure 19-6.

19.2.1 Additional WAP service installation

Browse WAP service

To add an additional browse WAP service, on the Resources tab right-click the **WAP proxy** to which you want to add a WAP service, then click **Add -> WAP service > Browse**.



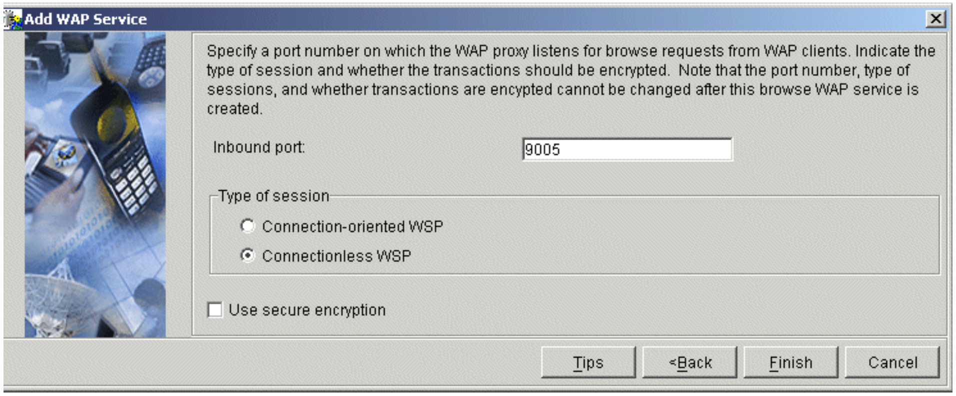
Add WAP Service

Connection Manager: cn=WAP proxy,cn=m10df5af,o=ibm,c=us

Description: WAP service

Figure 19-7 Service description

Give a descriptive name for the service. See Figure 19-7.


Add WAP Service

Specify a port number on which the WAP proxy listens for browse requests from WAP clients. Indicate the type of session and whether the transactions should be encrypted. Note that the port number, type of sessions, and whether transactions are encypted cannot be changed after this browse WAP service is created.

Inbound port: 9005

Type of session

○ Connection-oriented WSP

● Connectionless WSP

□ Use secure encryption

Figure 19-8 Inbound port, protocol and security settings

Insert a port number for inbound WAP traffic and choose the appropriate WSP connection type either **connection-oriented** or **connectionless**. If needed, you can enable **Use secure encryption** as well. See Figure 19-8.

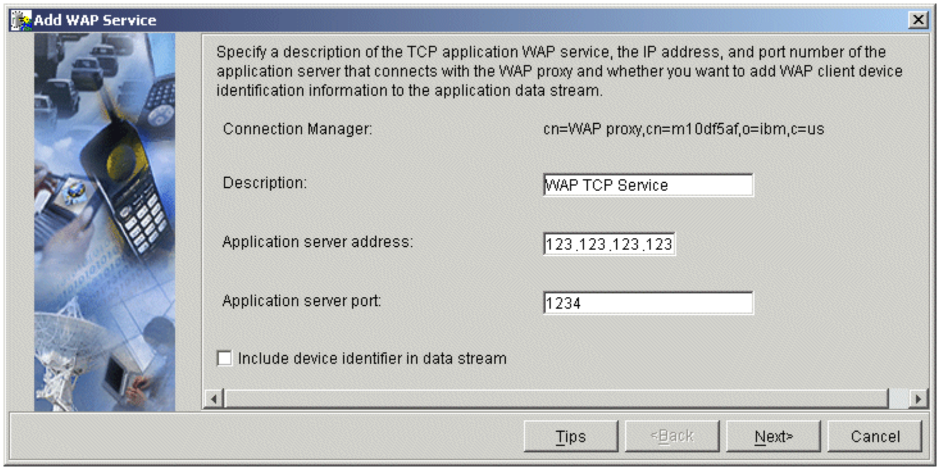
For more information about WAP service protocols, see “There are four types of WAP service protocols:” on page 416.

Note: Do not use predefined ports 9200, 9201, 9202, or 9203 for inbound ports. In addition, inbound ports, session types, and encryption settings cannot be changed after the Browse WAP service is created.

TCP application WAP service

The TCP application WAP service is a generic connectionless service that provides a conversion between TCP and WAP protocols. When the WAP proxy receives data on a specific port, it opens TCP socket connection to a specified application server, and passes data sent by the WAP client to the application server. When data is returned, it is converted back to WAP protocol and returned back to the WAP client.

To add an additional browse WAP service, on the Resources tab right-click the **WAP proxy** to which you want to add a WAP service, then click **Add -> WAP service -> TCP Application**.



Add WAP Service

Specify a description of the TCP application WAP service, the IP address, and port number of the application server that connects with the WAP proxy and whether you want to add WAP client device identification information to the application data stream.

Connection Manager: cn=WAP proxy,cn=m10df5af,o=ibm,c=us

Description: WAP TCP Service

Application server address: 123.123.123.123

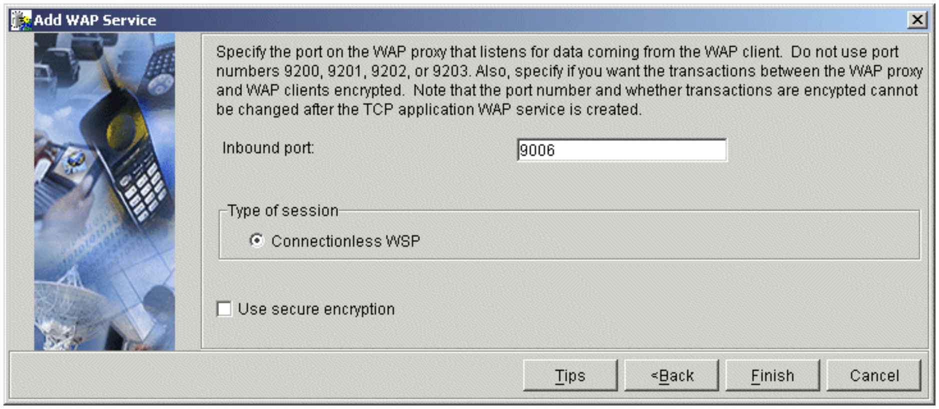
Application server port: 1234

□ Include device identifier in data stream

Figure 19-9 Service description

Give a descriptive name for the service. Enter an IP address and port number of the application server used. You can also include a device identifier in the header of the TCP data stream. See Figure 19-9.

> **Note:** If you choose to include a device identifier in data stream, the application receiving the data stream is responsible to interpret the header.



Add WAP Service

Specify the port on the WAP proxy that listens for data coming from the WAP client. Do not use port numbers 9200, 9201, 9202, or 9203. Also, specify if you want the transactions between the WAP proxy and WAP clients encrypted. Note that the port number and whether transactions are encypted cannot be changed after the TCP application WAP service is created.

Inbound port: 9006

Type of session

● Connectionless WSP

□ Use secure encryption

Figure 19-10 Inbound port and security

Insert a port number for inbound WAP traffic and choose appropriate WSP connection types (only connectionless is supported). If needed, you can enable **Use secure encryption** as well. See Figure 19-10.

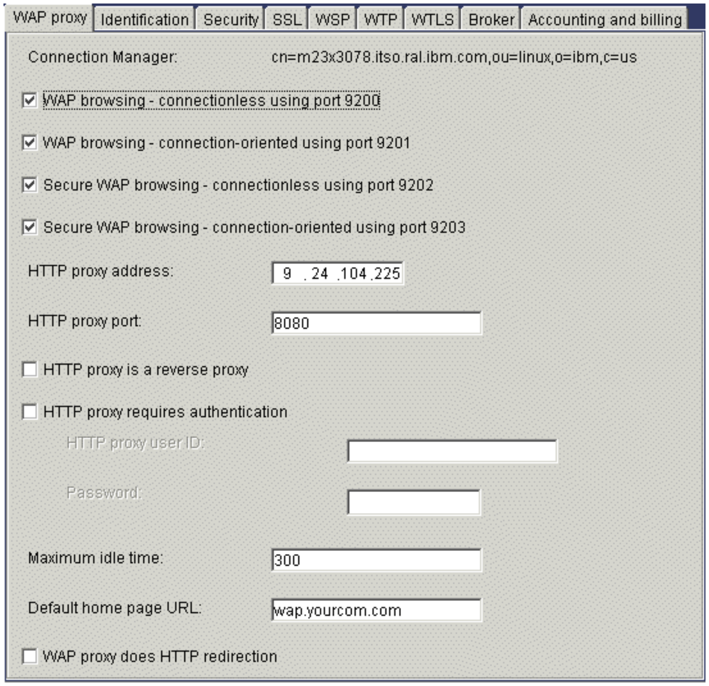
Note: Do not use predefined ports 9200, 9201, 9202, or 9203 for inbound ports.

Note: Inbound port and encryption settings cannot be changed after the TCP Application WAP service is created.

19.3 Configuration

Here we discuss some important settings that were not configured during installation.

19.3.1 WAP proxy



Connection Manager: cn=m23x3078.itso.ral.ibm.com,ou=linux,o=ibm,c=us

☑ WAP browsing - connectionless using port 9200

☑ WAP browsing - connection-oriented using port 9201

☑ Secure WAP browsing - connectionless using port 9202

☑ Secure WAP browsing - connection-oriented using port 9203

HTTP proxy address: 9 . 24 .104 .225

HTTP proxy port: 8080

☐ HTTP proxy is a reverse proxy

☐ HTTP proxy requires authentication

HTTP proxy user ID:

Password:

Maximum idle time: 300

Default home page URL: wap.yourcom.com

☐ WAP proxy does HTTP redirection

Figure 19-11 WAP proxy configuration

HTTP proxy considerations

In Connection Manager all requests from clients are always forwarded to an HTTP proxy. This is done because HTTP proxies are built to handle HTTP traffic and errors. This also frees WAP proxy to do its own job instead of wasting system resources to manage HTTP connections.

To set up a HTTP proxy, enter an IP address and a port number of the HTTP proxy. If the proxy requires authentication, enter a valid user ID and password. If a proxy is configured as a reverse proxy, check the check box.

There are three main scenarios, which we explain here:

- WAP proxy with forward proxy
- WAP proxy with reverse proxy
- WAP proxy with forward and reverse proxy

WAP proxy with forward HTTP proxy

This is a basic scenario for the HTTP proxy. Proxy is configured as a forward proxy, so all traffic flows through the proxy to the final destination (Figure 19-12). No extra configuration is required in proxy.

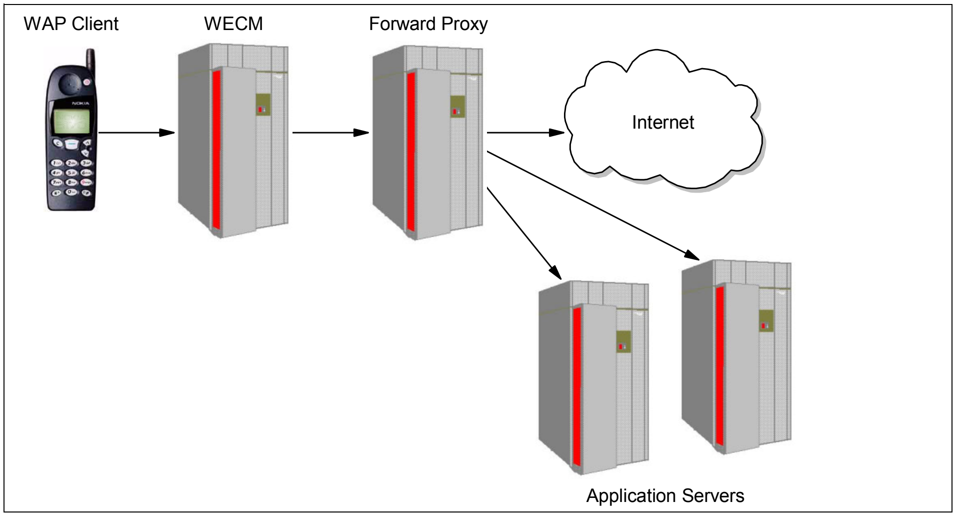


Figure 19-12 WAP proxy with forward HTTP proxy

This configuration allows clients to access any host accessible by HTTP proxy, for example, browsing the Internet.

WAP proxy with reverse HTTP proxy

In this scenario, proxy is configured as a reverse proxy. This means that clients can directly connect only to the proxy (Figure 19-13). This scenario requires extra configuration in proxy.

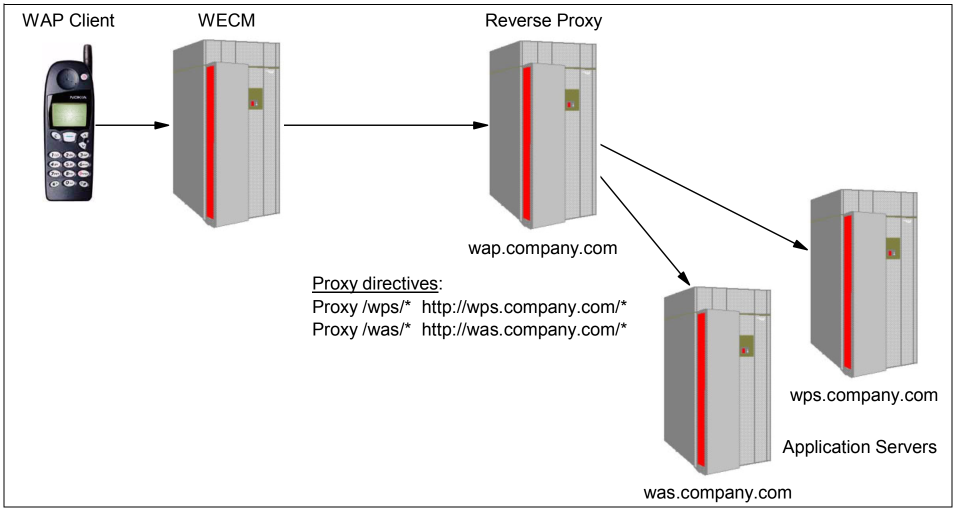


Figure 19-13 WAP proxy with reverse HTTP proxy

In our example, there are two proxy directives:

```
/wps/* http://wps.company.com
/was/* http://was.company.com
```

So, when a client connects to `http://wap.company.com/wps/portal` it is directed by proxy to: `http://wps.company.com/wps/portal`

This configuration is usually used when WAP clients are only accessing the corporate intranet, and it has the same advantages that a normal reverse proxy implementation has.

Note: When using this scenario, the only host that is directly accessible is the reverse HTTP proxy.

WAP proxy with forward and reverse HTTP proxy

This configuration has all advantages from both forward and reverse proxy configurations. It allows clients to directly access the Internet using forward proxy, and still provides reverse proxy protection of corporate intranets (Figure 19-14).

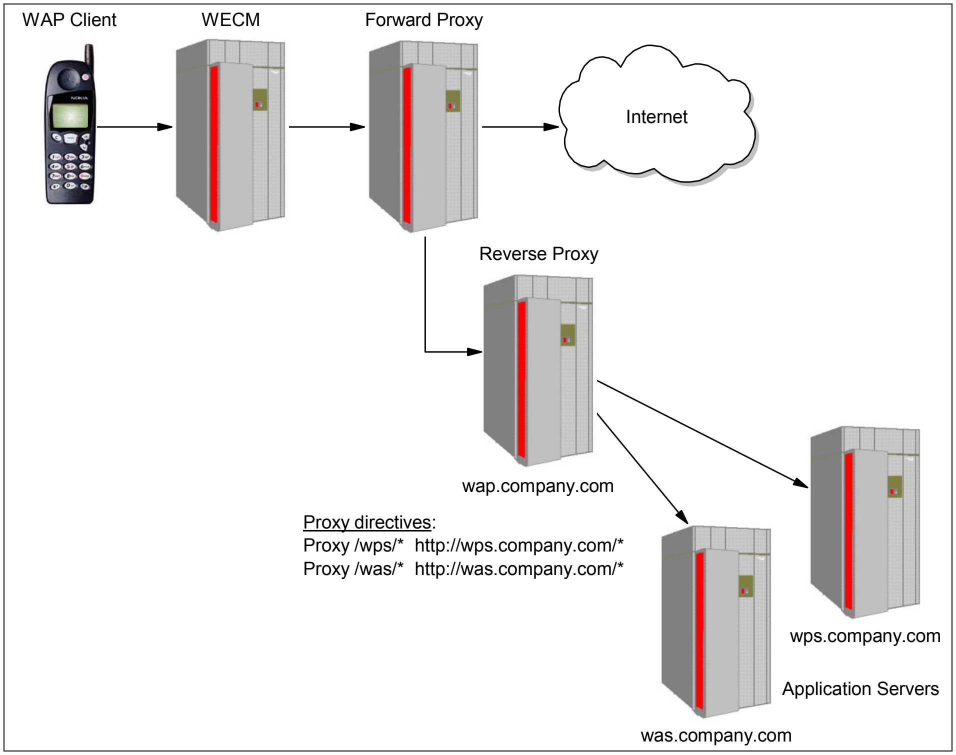


Figure 19-14 WAP proxy with forward and reverse HTTP proxy

HTTP redirection

You can select whether WAP proxy does HTTP redirection or not.

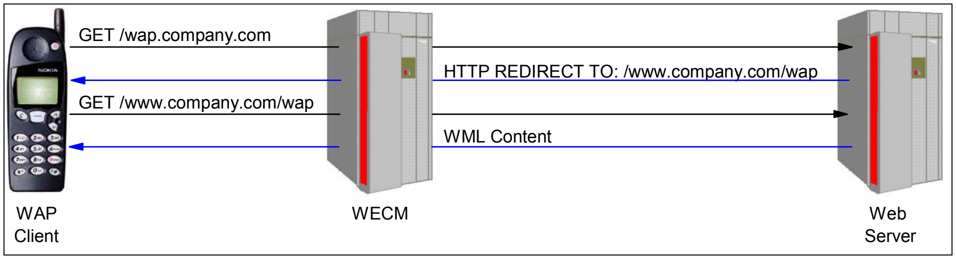


Figure 19-15 Proxy HTTP redirection disabled

In first scenario (Figure 19-15), WAP proxy does not do HTTP redirection. HTTP redirect is forwarded to the client device, which then does a new request to a given address.

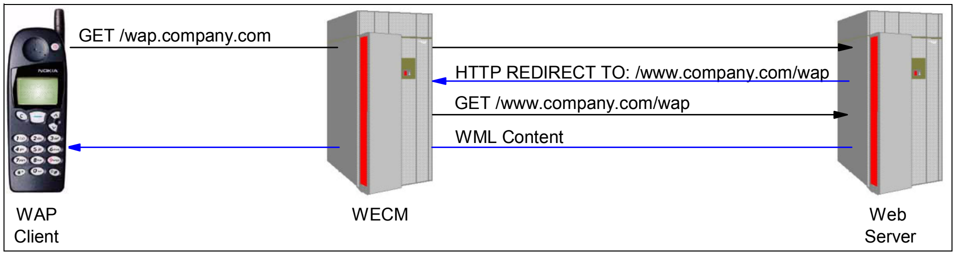


Figure 19-16 Proxy HTTP redirection enabled

In the second scenario (Figure 19-16) WAP proxy is configured to do HTTP redirection. WAP proxy notices that an HTTP redirection is returned so it will do the new request to a given address. When WAP proxy returns the content to client device, it looks like it came from an originally requested address. In this scenario, traffic between the WAP client and WAP proxy is reduced. A drawback is that if there are relative URLs used in the WML deck; they will point to the wrong place. For example, first the WAP device requests a page from `wap.company.com`, but actually the page returned is from `www.company.com/wap`, however, the client is not aware of that. Now, if there is a link in the returned page that points, for example, to `news.wml`, the request page is `wap.company.com/news.wml` when it was supposed to be `www.company.com/wap/news.wml`. Usage of absolute URLs will avoid the problem but that increases the size of the links, therefore decreases the size of the actual content that can be fitted on the page.

19.4 WAP proxy message book utility

The message book utility provides a set of pre-encoded WML documents that present HTTP error codes. When a Web server returns an HTTP error code, WAP proxy sends a corresponding WML document to the WAP client.

WML documents are installed automatically with the WAP proxy, and are located in the directory: `/usr/lpp/wireless/wap/messages/`*<lang>*/ (AIX) or `/opt/wireless/wap/data/`*<lang>*/ (Linux and Solaris), where *<lang>* is the language code sent by WAP client. The default language is English.

WML documents can be customized for your environment. WML documents are named as: `msg`*<code>*`.wml`, where *<code>* is number of HTTP error code, for example WML document for HTTP error 404 (page not found) is in file named `msg404.wml`.(See Example 19-1.) If no file is found for a given error code, then `msgdefault.wml` is used.

Example 19-1 msg404.wml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
  <card title="Error 404">
    <p>
      <do type="prev" name="back"><prev/></do>
       The requested page could not be found.
    </p>
  </card>

</wml>
```

For a complete list of the error codes, see Table 19-2.

Table 19-2 Message book utility.

Error code	File name	Text in WML document
400	msg400.wml	The WAP proxy could not understand you request.
401	msg401.wml	You need authorization to access this document.
403	msg403.wml	You are not authorized to access this WAP proxy.
404	msg404.wml	The requested page could not be found.
406	msg406.wml	The response cannot be displayed by this device.
407	msg407.wml	User authentication is required to access the WAP proxy.
413	msg413.wml	The document requested was too large to be sent to this device.
500	msg500.wml	The WAP proxy encountered an internal error while processing your request.

Error code	File name	Text in WML document
501	msg501.wml	The WAP proxy does not support the functionality to process your request.
502	msg502.wml	The WAP proxy encountered an error while processing your message.
503	msg503.wml	The WAP proxy service is temporarily unavailable.
504	msg504.wml	The WAP proxy timed out while retrieving your document.
1001	msg1001.wml	The WAP proxy could not decode the request headers.
1002	msg1002.wml	There was an error encoding the response.
1003	msg1003.wml	There was an error encoding the response headers.
1004	msg1004.wml	You are not allowed to access a secure Web site (HTTPS) without a secure WAP connections.
1005	msg1005.wml	This request has been cancelled by the gateway. Your account has timed out or another client has logged on with the same user ID.
1006	msg1006.wml	This request cannot be processed by the WAP proxy. You are already logged onto the gateway using another device. Please try again later.
All other codes	msgdefault.wml	The WAP proxy could not process your request.

19.5 WML transcoding and fragmentation

This section presents an overview of transcoding and fragmentation used to support WAP devices.

Transcoding

Most of the Web sites of the world are not WAP enabled, so they do not have any content in a WML format. Transcoding technology allows WAP clients to browse HTML content also by converting HTML files to WML. Also, pictures are converted to a wbmp format, which is the only supported format of WAP devices. This dramatically increases the number of the WAP browsable sites.

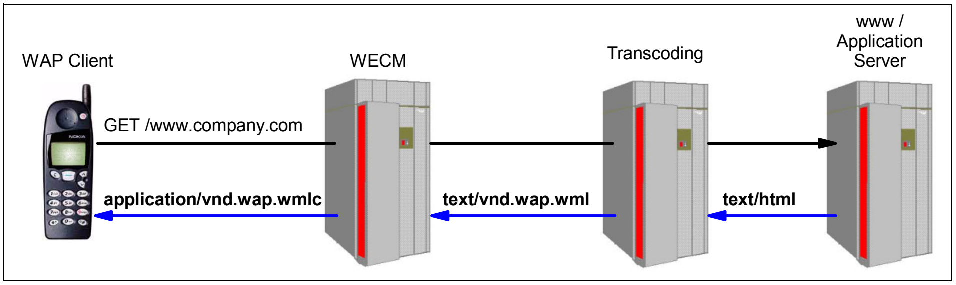


Figure 19-17 Transcoding technology

Fragmentation

WAP devices are usually devices with a limited capacity of memory to store WML decks. If the incoming deck is larger than the device supports, the deck is not displayed. This can be avoided by fragmenting content to smaller pieces, and then sending them to a device one at a time. Each part has a link to parts before and after it (if one exists). That way you can browse through the document easily.

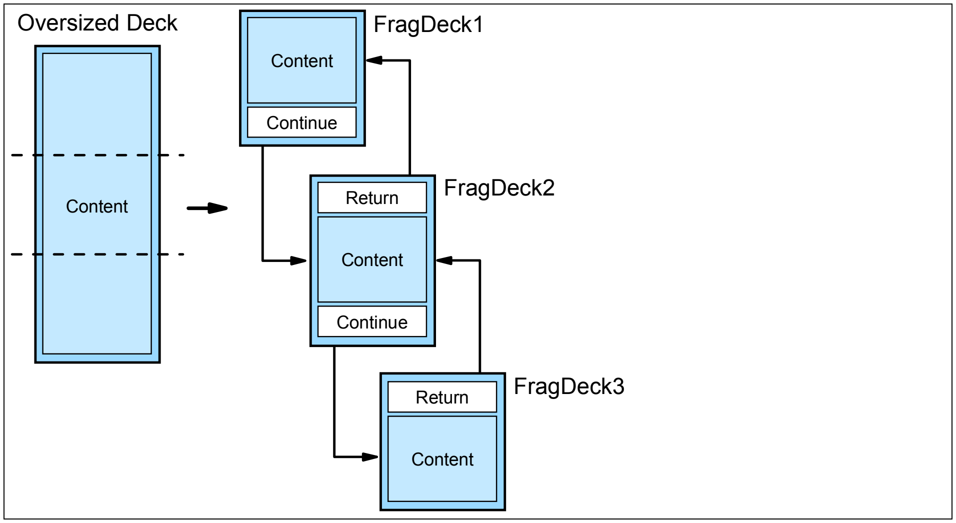


Figure 19-18 WML deck fragmentation

19.5.1 WebSphere Transcoding Publisher

Connection Manager does not provide WML transcoding nor fragmentation. To use transcoding and fragmentation, WebSphere Transcoding Publisher (WTP) is required.

There are several ways to use WTP with Connection Manager:

- As a standalone proxy, where both forward and reverse proxy configurations are supported
- As a plugin to the WebSphere Edge Caching proxy, formerly named as WebTraffic Express (WTE)
- As a plug-in to the WebSphere Application Server (WAS)
- As a part of the WebSphere Everyplace Access server (WEA)

The most common scenario is to use the WTP as a plugin to the WebSphere Edge Caching proxy. This combination provides transcoding and fragmentation, and also a competent caching. This reduces traffic between proxy and Web servers because most often the requested pages come from the cache.

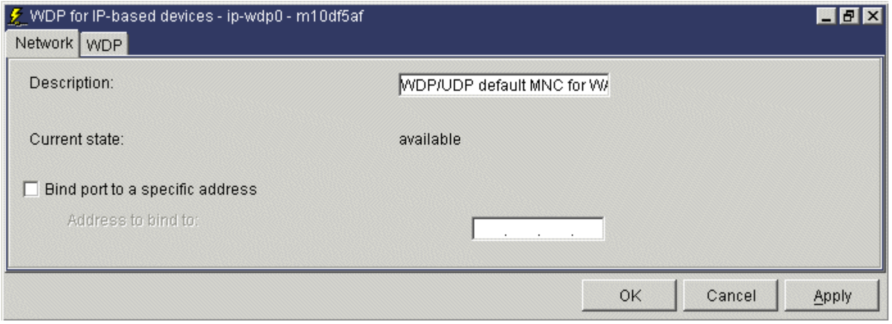
19.6 Mobile Network Connection (MNC) configuration

In this section we include a Mobile Network Connection (MNC) sample configuration for WAP proxy support.

19.6.1 IP Wireless Datagram Protocol configuration (ip-wdp)

IP/UDP bearer adapter using Wireless Datagram Protocol is a default protocol for the WAP proxy, and it is automatically installed with the Connection Manager.

To configure ip-wdp using Gatekeeper, on the Resources tab expand the **WAP proxy** that you want to configure, right-click **ip-wdp0**, then click **Properties.**



WDP for IP-based devices - ip-wdp0 - m10df5af

Description: WDP/UDP default MNC for W/

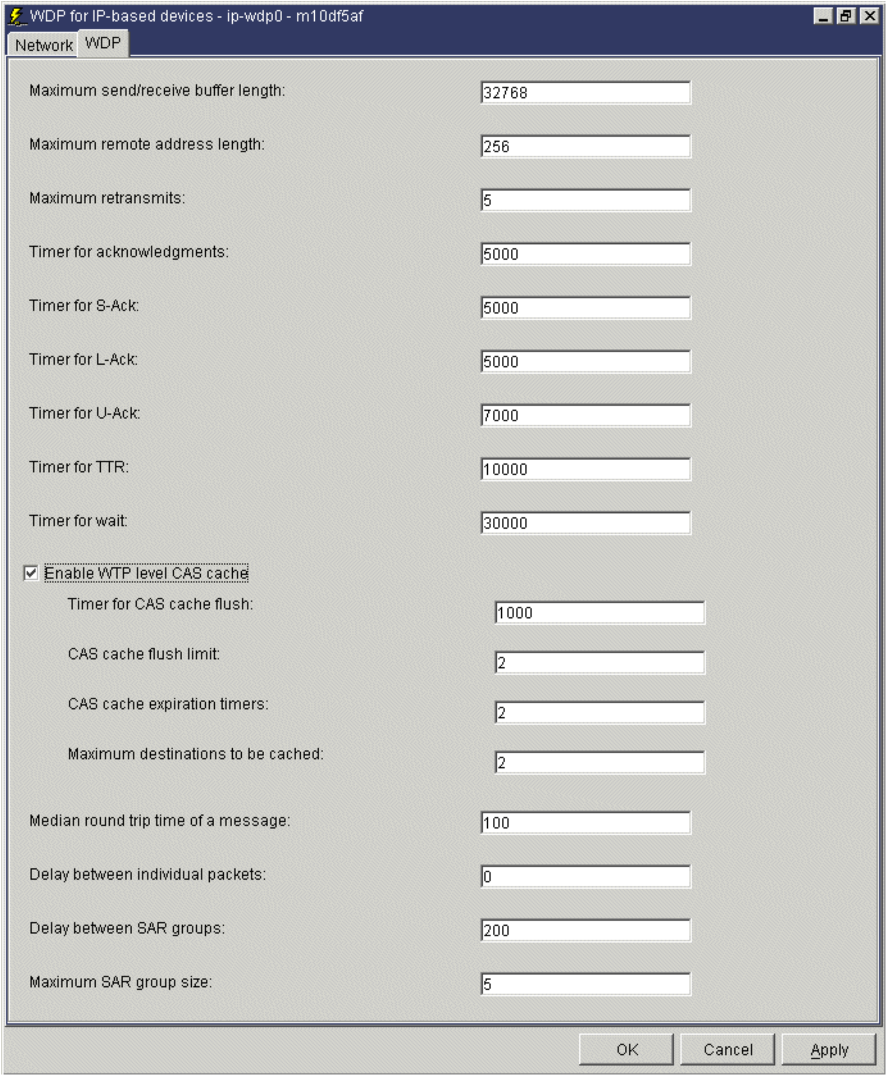
Current state: available

□ Bind port to a specific address

Address to bind to: . . .

Figure 19-19 WDP Network settings

Enter descriptive name for MNC. You can bind the MNC to use a specific IP address so that traffic is always routed from same address, even if another interface card is actually used to route packets. See Figure 19-19.



Maximum send/receive buffer length: 32768

Maximum remote address length: 256

Maximum retransmits: 5

Timer for acknowledgments: 5000

Timer for S-Ack: 5000

Timer for L-Ack: 5000

Timer for U-Ack: 7000

Timer for TTR: 10000

Timer for wait: 30000

☑ Enable WTP level CAS cache

Timer for CAS cache flush: 1000

CAS cache flush limit: 2

CAS cache expiration timers: 2

Maximum destinations to be cached: 2

Median round trip time of a message: 100

Delay between individual packets: 0

Delay between SAR groups: 200

Maximum SAR group size: 5

Figure 19-20 WDP settings

From this tab, you can adjust WDP protocol specific length and timer settings. You can also enable WTP level CAS caching. By default, WTP level cache is disabled.

20

WAP security

This chapter describes how to set up a secure environment for WAP devices. The topics covered in this chapter are:

- Secure end-to-end communication between WAP devices and back-end application servers
- Configuration of WTLS certificates
- Installation of SSL certificates
- WAP client identification

20.1 Overview

WebSphere Everyplace Connection Manager can be set up to secure all communication between WAP devices and application servers. Access to Connection Manager can also be controlled so that only authenticated users (or devices) are authorized to connect to back-end applications.

WAP devices use the Wireless Session Protocol (WSP) to access the WebSphere Everyplace Connection Manager. Connection Manager does a protocol conversion into the Hypertext Transfer Protocol (HTTP), and the request is then sent to the application server through a HTTP proxy. That means, two network connections are established that must be secured in order to achieve a secure end-to-end communication. In the case where a HTTP reverse proxy is used, the HTTP connection from the Connection Manager ends at the proxy, which then establish a connection to the secure application server.

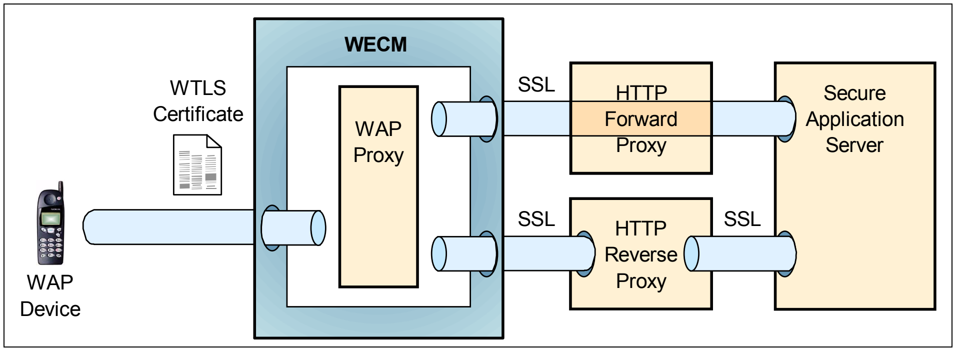


Figure 20-1 WAP security overview

The connection between WAP devices and Connection Manager can be secured using the Wireless Transport Layer Security protocol (WTLS). The Secure Socket Layer protocol (SSL) is used to establish a secure connection from the WAP proxy to back-end application servers.

20.1.1 Wireless Transport Layer Security

WTLS as part of the Wireless Application Protocol (WAP) is used to encrypt all communication between WAP clients and Connection Manager. Connectionless as well as connection-oriented links are supported for secure WAP browsing. Connection Manager can also be configured to use secure transactions for TCP applications.

WebSphere Everyplace Connection Manager supports various protocols for key exchange and encryption. Key exchange can be configured to use anonymous and certificate-based algorithms. In the case of a certificate-based algorithm, the Connection Manager presents its certificate to the WAP client during the first request of a session in order to prove its own identity.

The following key exchange suite are supported:

- RSA anonymous (512, 768 and 1024 bit key strength)
- RSA certificate-based (512, 768 and 1024 bits key strength)
- Diffie-Hellman anonymous (512, 768 and 1024 bit key strength)
- Elliptic Curve Diffie-Hellman anonymous (113 and 131 bit key strength)

The following cipher suites are supported for encryption:

- DES (56 bit key strength)
- Triple-DES (168 bit key strength)
- RC5 (40, 56 and 128 bit key strength)

20.1.2 Secure Sockets Layer

WebSphere Everyplace Connection Manager performs a protocol conversion for all WAP requests into the HTTP protocol. The HTTP requests and the corresponding responses from the back-end application servers can be encrypted using the Secure Sockets Layer (SSL) protocol. Connection Manager supports the following cipher algorithms:

- DES (56 bit key strength)
- Triple-DES (168 bit key strength)
- RC5 (40, 56 and 128 bit key strength)

The use of SSL to secure the communication to a back-end application server can be restricted to connections where the connection between the WAP device and Connection Manager is secured by WTLS.

In order to establish a SSL connection between Connection Manager and a application server, Connection Manager must be able to validate the application server’s certificate.

The default key database contains the root certificates of various commercial Certificate Authorities such as Verisign, Thawte, and others. If the application server uses a certificate signed by one of these Certificate Authorities, the key database can be used as is. If the certificate of the application server is signed by another CA, or uses a self-signed certificate, the IBM Key Management utility that ships with the Connection Manager can be used to create a new or update the existing key database.

20.1.3 WAP client identification

Connection Manager provides several mechanisms to uniquely identify or authenticate the WAP client. This allows you to restrict access to the WAP proxy to a closed user group. The following methods are supported to validate a WAP request:

- User ID/password based authentication
- Device resolver
- External account lookup program

Only one mechanism can be configured per WAP proxy. The following picture shows an overview of all validation methods, and how they interact with the WAP proxy.

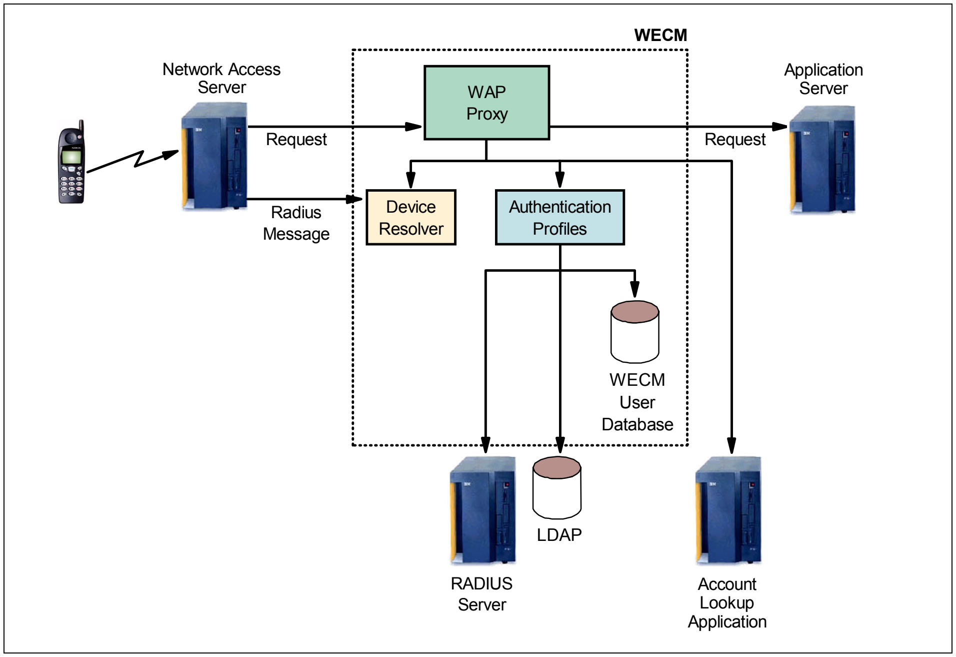


Figure 20-2 WAP client identification methods

The user ID/password based authentication uses basic HTTP proxy authentication to challenge the WAP client for the user ID and password. The user authentication credentials in the WAP request must match those defined for the user at the Connection Manager before the request is accepted. The WAP proxy can be configured to use one of the authentication profiles defined in the Connection Manager.

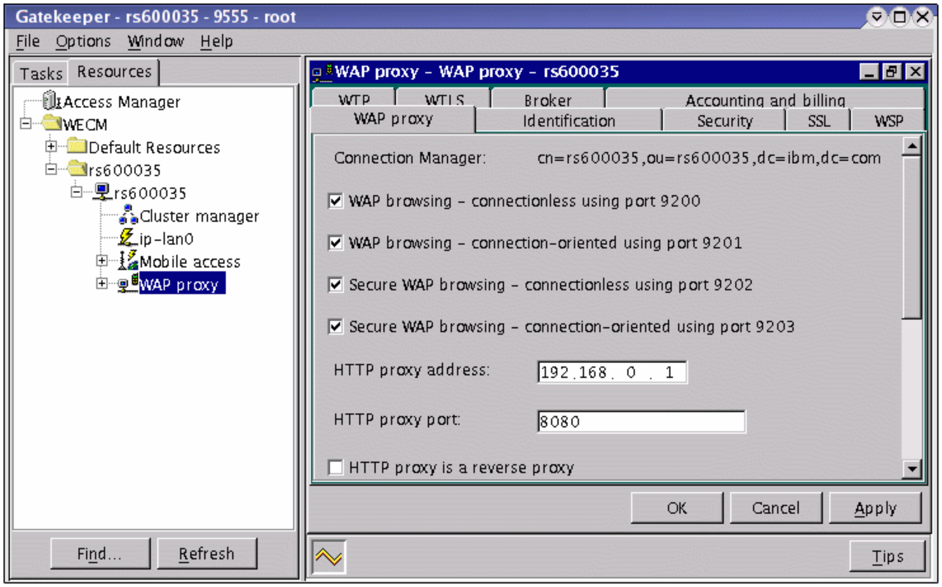
The device resolver and external account lookup program are not methods of validation or authentication, but map unique identification characteristics of a WAP client. Validation or authentication must occur elsewhere in the environment, typically at the network access server.

When using the device resolver, a RADIUS message containing unique device identification must be received form a network access server (NAS) before the WAP request is accepted by the Connection Manager. The unique device information can be in the form of a WAP device phone number or MSISDN number, user ID, or other device specific information.

The external account lookup program can be used to verify the user's identification. The WAP proxy forwards the WAP request containing an identifier of the WAP client to the account lookup program. The identifier depends on the WAP bearer type and can be any string, but typically it is a dotted-decimal IP address. The program then matches the identifier to a unique user ID, and returns it to the WAP proxy. The returned user ID must match a WAP user defined in Connection Manager.

20.2 Configuration

During the installation of the WAP proxy, you can choose to include secure WAP browsing services, connection oriented, and connectionless; for details see Chapter 19, "WAP gateway" on page 415. The services are then listed in the tree structure on the left hand site under WAP proxy. If you want to enable or disable a service, open WAP proxy in Gatekeeper and select or deselect the appropriate check box in the WAP proxy tab. See Figure 20-3 for an example.



Connection Manager: cn=rs600035,ou=rs600035,dc=ibm,dc=com

☑ WAP browsing - connectionless using port 9200

☑ WAP browsing - connection-oriented using port 9201

☑ Secure WAP browsing - connectionless using port 9202

☑ Secure WAP browsing - connection-oriented using port 9203

HTTP proxy address: 192 . 168 . 0 . 1

HTTP proxy port: 8080

☐ HTTP proxy is a reverse proxy

Figure 20-3 WAP proxy configuration

WebSphere Everyplace Connection Manager can be configured to secure WAP TCP services using WTLS. Select **Use secure encryption** during the creating of the TCP service; for details see Chapter 19, “WAP gateway” on page 415. Security can only be enabled during the creation of a WAP TCP service. If you forget to enable the security, then you have to delete the TCP service and recreate it.

20.2.1 Enable WAP client identification

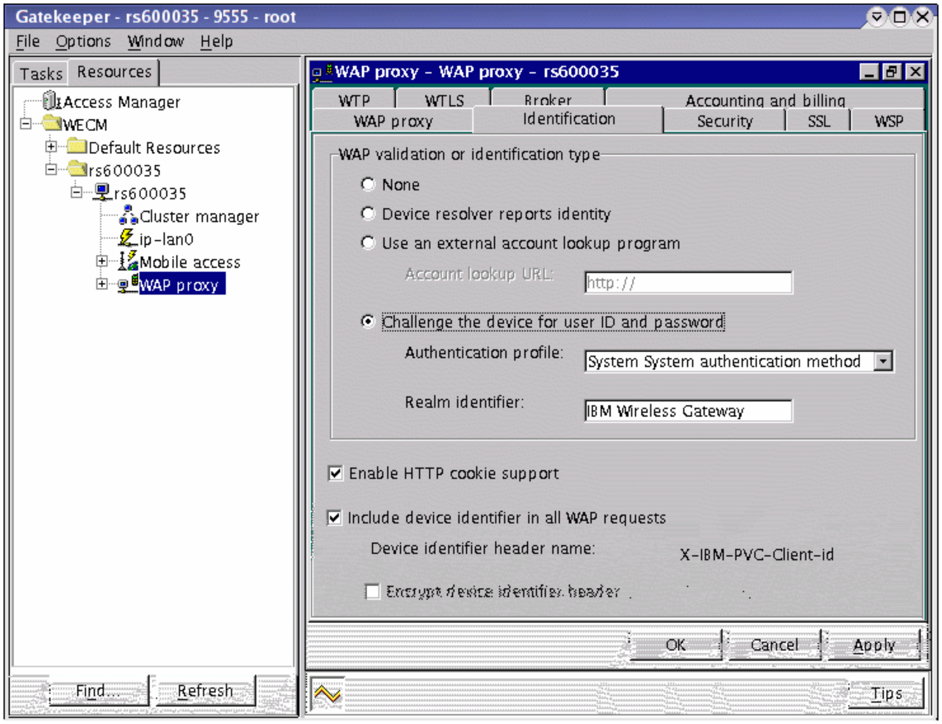
Authentication

The Connection Manager can be configured to restrict access to the WAP proxy to authenticated user. The WAP proxy uses HTTP proxy authentication to request the WAP client user ID and password.

When a WAP client connects to the Connection Manager for the first time during a session, the WAP proxy returns a HTTP proxy authentication challenge to the WAP client. The client device must send the correct user ID and password before it is allowed to access the WAP proxy. After a successful login, the WAP client stores the credentials for the duration of the session and sends them automatically on all subsequent requests to the WAP proxy.

To enable user ID and password based authentication, select the **Identification** tab of the WAP proxy, and select **Challenge the device for user ID and**

password. Select the authentication profile you want to use. The WAP proxy provides an authentication profile named system authentication method, which uses the credentials of the user defined in Connection Manager. You can also select one of the authentication profiles defined in Connection Manager, except certificate-based authentication. The **Realm identifier** specifies the text that is presented to the user during an authentication challenge.



WAP validation or identification type

○ None

○ Device resolver reports identity

○ Use an external account lookup program

Account lookup URL: http://

● Challenge the device for user ID and password

Authentication profile: System System authentication method

Realm identifier: IBM Wireless Gateway

☑ Enable HTTP cookie support

☑ Include device identifier in all WAP requests

Device identifier header name: X-IBM-PVC-Client-id

□ Encrypt device identifier header

Figure 20-4 Configure WAP client identification

Note: Certificate-based authentication profiles are not valid for use with the WAP proxy.

Account lookup program

To enable an account lookup, select **Use an external accounting lookup program** and enter the URL of the account lookup application.

When the external accounting lookup option is enabled, the WAP proxy sends an account lookup request, which is a HTTP 1.1 GET request to the given address using following request format:

```
http://<server>/<uri>?lookupaddress=<address>
```

where <server> is the host name or IP address of the application server, <uri> is the URI of a CGI script or servlet that performs the lookup, and <address> is the address of the account to lookup.

Note: lookupaddress=<address> is automatically appended to the end of the provided application server URL.

The format of the account address depends on the WAP bearer type that is used by the client. It can be a dotted decimal IP address or phone number, for example.

The content type of the response that is returned by the account lookup program should be plain text, and content should have the following format:

```
WG_LOOKUP: <addr>=<accountid>
```

Where <addr> is the queried address and <accountid> is the account ID of the Connection Manager user. If the account cannot be resolved, then <accountid> must be empty.

If the account cannot be resolved, or the account does not have rights to use WAP proxy, an HTTP error 403 (forbidden) is returned to WAP device.

In the following sample scenario, Connection Manager is used as WAP gateway to serve multiple customers. All customers are using GPRS as their bearer, and they have their own access points. There is one account per customer, which is mapped to the IP address of the access point of each customer; see Figure 20-5.

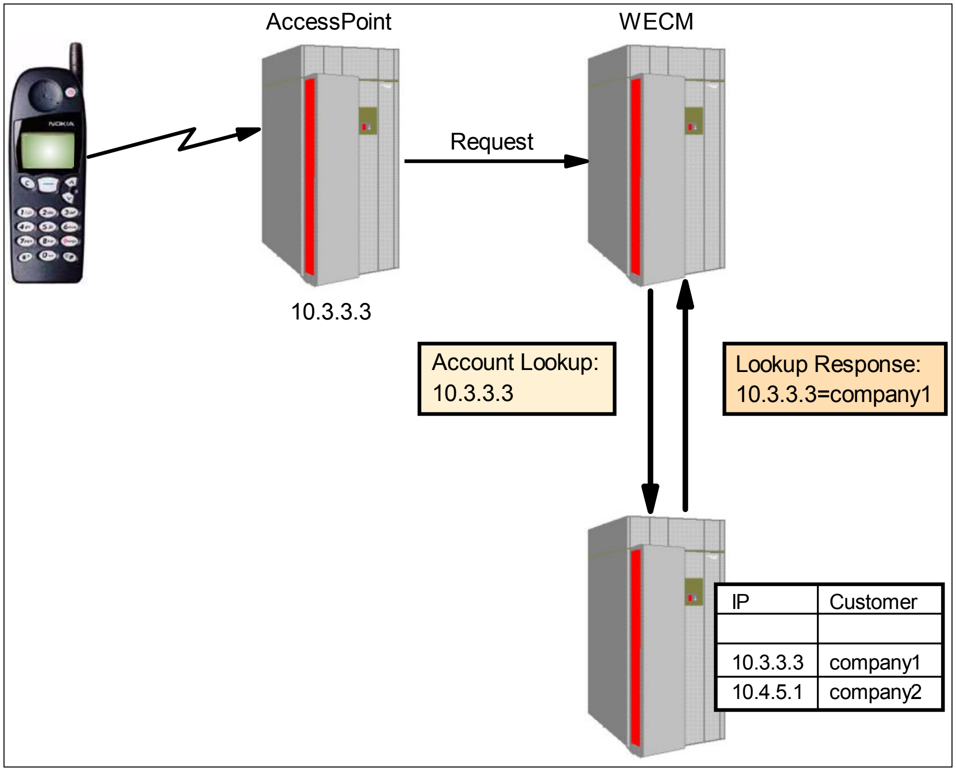


Figure 20-5 Sample external account lookup scenario

A servlet running on WebSphere Application Server is used to receive an account lookup request (Example 20-1).

Example 20-1 AccountLookup.java

```
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.*;

public void doGet(HttpServletRequest req, HttpServletResponse res)
   throws ServletException, IOException
{
   String address=req.getParameter("lookupaddress");
   String account="company1"; // Add your lookup code here
   String response="WG_LOOKUP: "+address+"="+account;

   res.setContentType("text/plain");
   res.getOutputStream().write(response.getBytes());
   }
```

```
}
```

When a customer is connecting to Connection Manager, a lookup request is sent to the lookup servlet using the following URL:

```
http://was.company.com/servlet/AccountLookup?lookupaddress=10.3.3.3
```

The complete HTTP request looks similar to Example 20-2.

Example 20-2 Account lookup HTTP request

```
GET /servlet/AccountLookup?lookupaddress=10.3.3.3 HTTP/1.1
Host: was.company.com
Accept: text/plain,text/html:q=0.8
Connection: close
```

The response that is returned by the servlet looks similar to:

```
WG_LOOKUP: 10.3.3.3=company1
```

Device resolver

The device resolver works in conjunction with a network access server, and is used to uniquely identify the WAP client device connecting to the WAP proxy. It does not provide validation or authentication. When using the device resolver, authentication must occur elsewhere in the environment, typically at the network access server.

When using the device resolver, a unique identifier of the device is sent from the network access server to the Connection Manager upon request. The identifier is sent in form of a RADIUS authentication or RADIUS accounting message, and must be sent before the Connection Manager accepts any requests from the device.

The device resolver can be configured to return RADIUS responses directly to the NAS, or it can be configured to act as a RADIUS proxy. Then, the device resolver forwards the RADIUS messages to other servers in the network, and returns the subsequent responses back to the NAS.

The type of RADIUS attribute that is used to uniquely identify the devices depends on the configuration of the network and the network access server. In most cases, the IP address does not uniquely identify a user because the NAS is usually configured to assign an IP address from a pool of addresses to a device. Thus, a device does not get the same IP address each time it connects to the network.

The RADIUS attribute type that is used to identify the client device can be configured in the device resolver. These attribute types are defined in RADIUS authentication RFC 2865 (http://www.ietf.org/rfc/rfc2865.txt) and RADIUS accounting RFC 2866 (http://www.ietf.org/rfc/rfc2866.txt). The identifier must be unique for the device and must be the same each time a particular device connects to the network. The device resolver supports the following RADIUS attribute types.

Table 20-1 RADIUS attribute types

RADIUS attribute type	RADIUS attribute name	Description
1	User-Name	Indicates the name of the user
31	Calling-Station-Id	Mobile phone number or MSISDN
Attribute type defined in RFC 2865 or RFC 2866		The attribute type must have a printable RADIUS attribute value (it cannot be binary). The attribute value must be unique for each phone.

If Network Address Translation (NAT) is used to translate the IP addresses between the network access server and the Connection Server, the use of the device resolver is not supported. The NAT server changes the source IP address to address assigned by the NAT server. Typically, the IP address assignment is dynamic so a particular device does not get the same IP address each time it connects to the network. The problem is that the RADIUS message contains the IP address that was assigned to the device by the NAS which does not match the IP address that was assigned by the NAT. The NAT simply modifies the source address of a request, and is not aware of the data that is sent.

Note: The use of the device resolver is not supported if network address translation is used between the network access server and the Connection Manager.

The following steps are involved in order to enable the WAP proxy to use the device resolver:

1. Configure the device resolver as a RADIUS server in NAS.
2. Add the device resolver in Connection Manager.
3. Enable the device resolver in the WAP proxy.

Network access server configuration

The network access server must be configured to send RADIUS authentication or accounting messages to the Connection Manager. The actual configuration of the NAS depends on the software that is used. The network access server sends device identification information to the Connection Manager in RADIUS messages, which must contain the following attributes:

- IP address

  The RADIUS must contain the RADIUS attribute type 8, which is the standard attribute type for the Framed-IP-Address.

- Unique device identifier

  The RADIUS message must contain an attribute type that uniquely identifies the WAP device. See Table 20-1 for all supported attribute types.

Please refer to the documentation of the network access server product for more information on how to configure it.

Add a device resolver

To add a device resolver, right-click on your **Connection Manager** and select **Add -> Device resolver**. The first panel lets you specify the name of the device resolver and a brief description. Click **Next** to go to the second panel and provide the following information:

- Network access server IP address

  The IP address of the network access server. The WAP proxy does not accept RADIUS messages that are not configured in to use the device resolver.

- RADIUS listen port

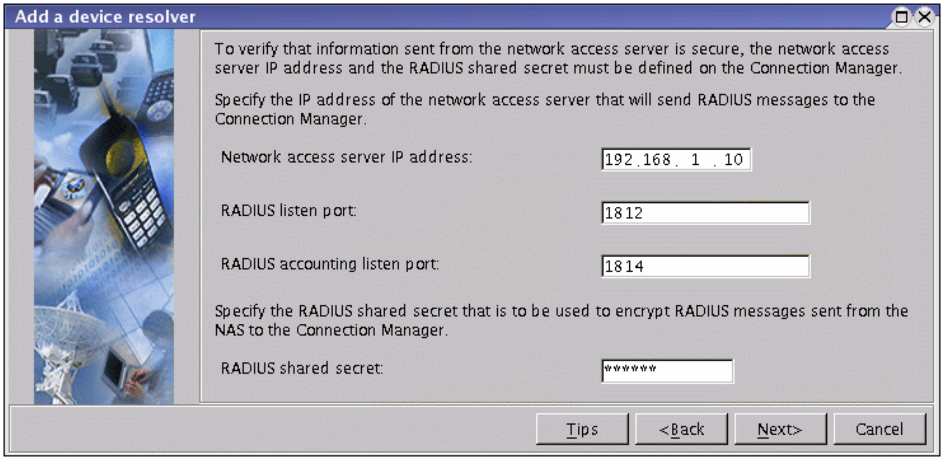
  The port on which the device resolver is listening for RADIUS messages

- RADIUS accounting listen port

  The port on which the device resolver is listening for RADIUS accounting messages

- RADIUS shared secret

  Specifies the shared secret that is used to encrypt the RADIUS messages. The shared secret must match the shared secret that is defined in NAS.



Add a device resolver

To verify that information sent from the network access server is secure, the network access server IP address and the RADIUS shared secret must be defined on the Connection Manager.

Specify the IP address of the network access server that will send RADIUS messages to the Connection Manager.

Network access server IP address: 192.168.1.10

RADIUS listen port: 1812

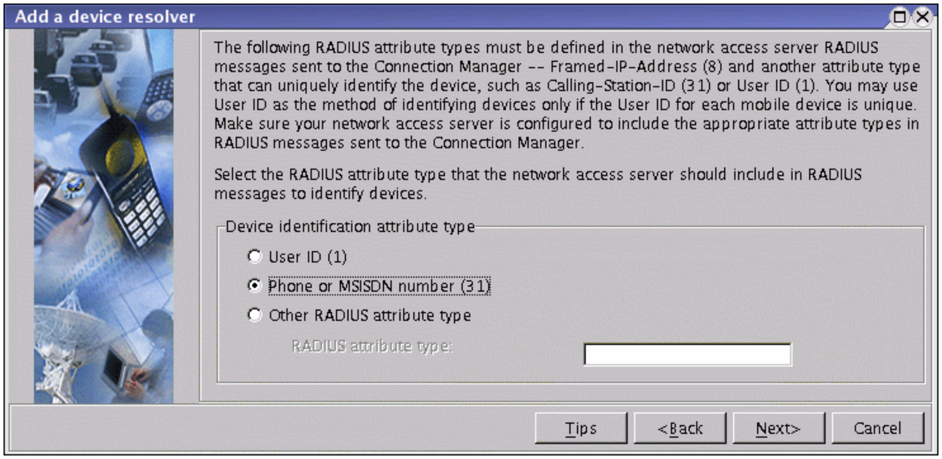
RADIUS accounting listen port: 1814

Specify the RADIUS shared secret that is to be used to encrypt RADIUS messages sent from the NAS to the Connection Manager.

RADIUS shared secret: *******

Figure 20-6 Device resolver: Network access server

The next panel configures the RADIUS attribute type that is used to uniquely identify the WAP device. See Table 20-1 on page 447 for the supported attribute types.



Add a device resolver

The following RADIUS attribute types must be defined in the network access server RADIUS messages sent to the Connection Manager -- Framed-IP-Address (8) and another attribute type that can uniquely identify the device, such as Calling-Station-ID (31) or User ID (1). You may use User ID as the method of identifying devices only if the User ID for each mobile device is unique. Make sure your network access server is configured to include the appropriate attribute types in RADIUS messages sent to the Connection Manager.

Select the RADIUS attribute type that the network access server should include in RADIUS messages to identify devices.

Device identification attribute type

○ User ID (1)

● Phone or MSISDN number (31)

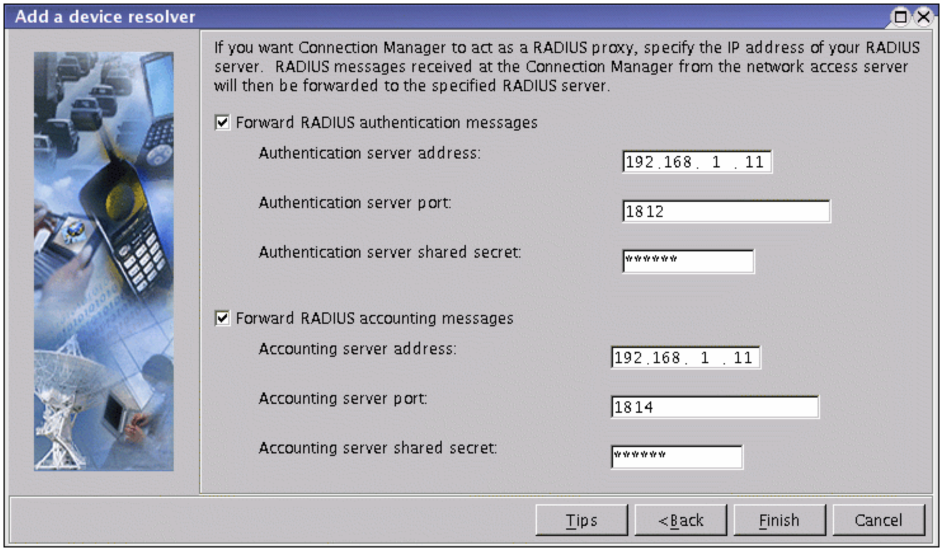
○ Other RADIUS attribute type

RADIUS attribute type:

Figure 20-7 Device resolver: RADIUS attribute type

In the last panel, the device resolver can be configured to forward the RADIUS authentication messages and/or RADIUS accounting messages to a RADIUS server. If enabled, the device resolver forwards the RADIUS message to the specified RADIUS server and returns the response from the RADIUS server to the network access server. If disabled, the device resolver returns a positive acknowledgement to any RADIUS authentication request.

Enter the IP address, the listening port, and the shared secret of the RADIUS authentication server and the RADIUS accounting server.


Add a device resolver

If you want Connection Manager to act as a RADIUS proxy, specify the IP address of your RADIUS server. RADIUS messages received at the Connection Manager from the network access server will then be forwarded to the specified RADIUS server.

☑ Forward RADIUS authentication messages

Authentication server address: 192 . 168 . 1 . 11

Authentication server port: 1812

Authentication server shared secret: *******

☑ Forward RADIUS accounting messages

Accounting server address: 192 . 168 . 1 . 11

Accounting server port: 1814

Accounting server shared secret: *******

Figure 20-8 Device resolver: Forward RADIUS messages

Configure WAP proxy to use the device resolver

To configure the WAP proxy to use a device resolver, open the WAP proxy and select the **Identification** tab. Select **Device resolver reports identity** (see Figure 20-4 for details).

There is an example of using the device resolver in Appendix B, “Simple device resolver implementation using NAS and RADIUS” on page 557.

20.2.2 Installation of a WTLS certificate

In order to configure the WAP proxy to use a WTLS server certificate to secure the connection between Connection Manager and WAP devices, use the tool wg_cert. It is a command line utility and available on the system where Connection Manager is installed.

The wg_cert tool will create your private key and a certificate signing request. The certificate signing request is then submitted to a Certificate Authority (CA), which will return your server certificate signed by the CA.

You can install only one WTLS server certificate at a time. If a certificate is already installed, and you choose to create a new one, then the installed certificate will be overwritten:

1. Make sure your are logged in as root. Create the private key and the certificate signing request by issuing the following command on a command line:

   `wg_cert -c`

2. Choose your key strength, which can be either 512, 768, or 1024 bit, and provide your country code, organization, Organizational Unit, and the server name. All values must be in X.500 notation and become part of the certificate request. For example, `cn=WECM5,ou=ITSO,o=IBM,c=US` where `cn` is the server name; `ou` is the Organizational Unit; `o` is the organization; and `c` is the country. See Example 20-3 for sample input and output of the utility.

Example 20-3 Create a certificate signing request

```
# wg_cert -c
Enter your Key Strength (512,768,1024):
1024
Enter your Country Code:
c=US
Enter your Organization:
o=IBM
Enter your Organization Unit:
ou=ITSO
Enter your Server Name:
cn=WECM5
WTLS certificate signing request successfully stored to file wtlscsr.p10
WTLS certificate (key ServerWTLSCerts/0a_csr) - store successful!
WTLS private key successfully stored to file wtlspri.asc
```

   After you have provided all values, two files are created in the current directory: wtlspri.asc contains your private key in encode format, and wtlscsr.p10 contains the certificate signing request in PKCS#10 format. Keep these files until you have stored the information you received from the Certificate Authority.

3. Submit to certificate signing request (wtlscr.p10) to a Certificate Authority of your choice through a Web browser. You can choose either binary or PEM Base64 format for the certificate. The CA will return two files to you: its own root certificate and your server certificate.

4. Transfer the files to the system where you created the certificate signing request.

5. Install your server certificate and the CA root certificate by entering the following command:

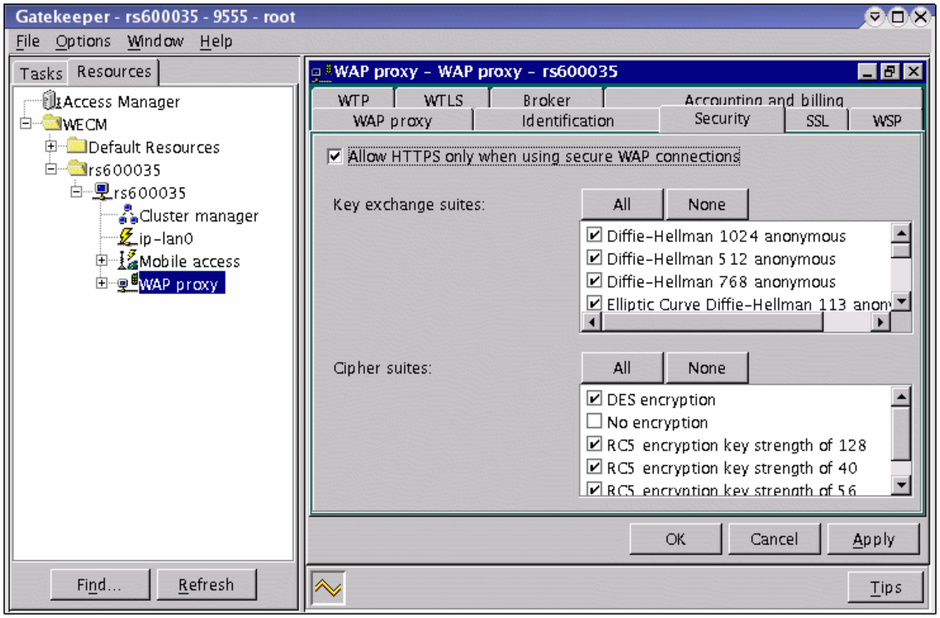
   `wg_cert -s wtlspri.asc <servercert> <rootcert>`

   where `<servercert>` is the filename of your server certificate and `<rootcert>` is the filename of the CA root certificate. See the following example:

Example 20-4 Install WAP server certificate

```
# wg_cert -s wtlspri.asc your-wtls-cert-class0.bin tc_wtls_cert_ca.bin
WTLS certificate (key ServerWTLSCerts/0a_priv) - store successful!
File tc_wtls_cert_ca.bin does not appear to be in PEM format - attempting to
read in as BER encoded binary data.
File your-wtls-cert-class0.bin does not appear to be in PEM format - attempting
to read in as BER encoded binary data.
WTLS certificate (key ServerWTLSCerts/0a) - store successful!
WTLS certificate (key ServerWTLSCerts/0b) - store successful!
You must restart the Connection Manager to incorporate your changes
```

6. Shut down and start up the Connection Manager to activate the changes.
7. Start the Gatekeeper and log on to the Connection Manager.
8. Select the **WAP proxy** and select the **Security** tab.
9. Select the key exchange and cipher suites that match the values for your certificate; see Figure 20-9 for an example.



☑ Allow HTTPS only when using secure WAP connections

Key exchange suites:
☑ Diffie-Hellman 1024 anonymous
☑ Diffie-Hellman 512 anonymous
☑ Diffie-Hellman 768 anonymous
☑ Elliptic Curve Diffie-Hellman 113 anon

Cipher suites:
☑ DES encryption
☐ No encryption
☑ RC5 encryption key strength of 128
☑ RC5 encryption key strength of 40
☑ RC5 encryption key strength of 56

Figure 20-9 WAP security configuration

All certificates can be deleted from the certificate store by using the following command (restart the Connection Manager to activate the changes):

```
wg_cert -d
```

To retrieve a previously created certificate, use the following command:

```
wg_cert -f
```

You can only retrieve certificates if you have previously stored them.

20.2.3 Manage SSL certificates

You can configure public key certificates to use SSL and enable a secure connection between the WAP proxy and secure application servers.

In order to establish a secure connection between Connection Manager and the application server, the application servers' SSL certificate must be installed in the WAP proxy key database.

The use of SSL can be restricted to secure WAP connections only. If this option is enabled in the WAP proxy (see Figure 20-9) then the Connection Manager allows SSL connections only if the WAP device is using a secure WTLS connection to access the WAP proxy.

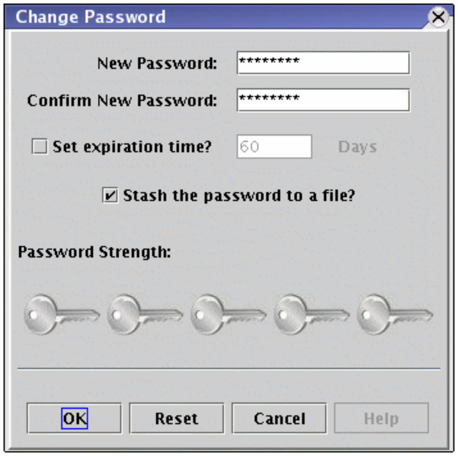
Manage the key database

WebSphere Everyplace Connection Manager ships with a default key database containing various certificates of common Certificate Authorities. The key database is located in the root directory of the Connection Manager installation, and its filename is wgated.trusted.kdb. The default stash password is trusted and stored in wgated.trusted.sth at the same location.

You can change the password or create a new key database using the wg_ikeyman utility that is part of Connection Manager. You have to set the environment variable JAVA_HOME in order to start wg_ikeyman. See the following example:

```
# export JAVA_HOME=/usr/java131
# wg_ikeyman
```

To change the password of your existing key database, open your key database and select **Key Database File -> Change Password**. Enter your new password and select **Stash the password to a file?**. This will then change the password and store it in a file. You can also stash the password to a file by selecting **Key Database File -> Stash Password**.



Change Password

New Password: ********

Confirm New Password: ********

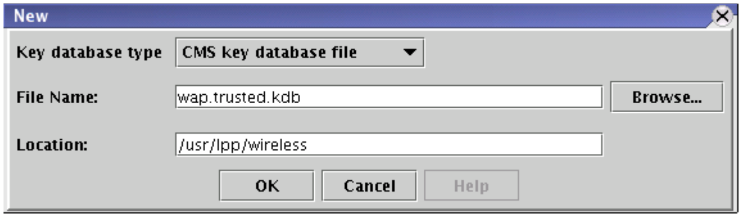
□ Set expiration time? 60 Days

☑ Stash the password to a file?

Password Strength:

Figure 20-10 Change key database password

To create a new key database, start the **wg_ikeyman** utility and select **Key Database File -> New**. A new window will open. Select **CMS key database file** as the key database type, and enter the filename and location of the key database.



New

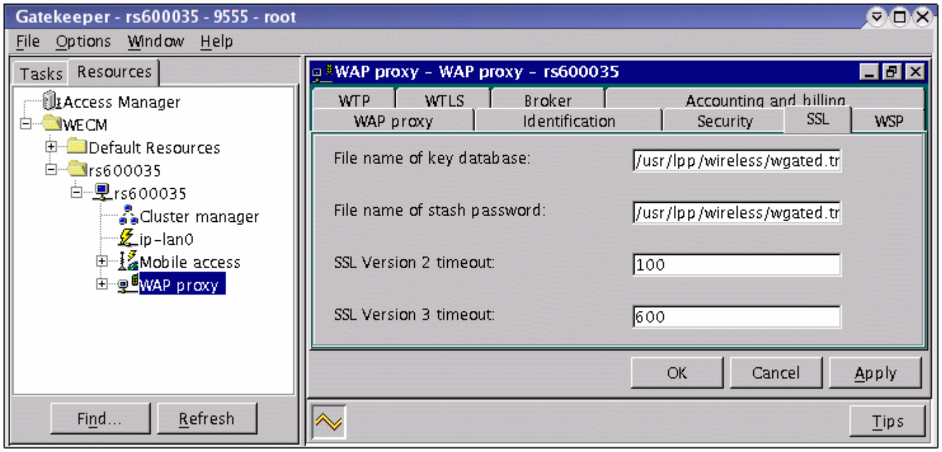
Key database type: CMS key database file

File Name: wap.trusted.kdb

Location: /usr/lpp/wireless

Figure 20-11 Create new key database

The file name and location of the key database and its corresponding stash password can be configured in the **SSL** tab of the WAP proxy settings.



File name of key database: /usr/lpp/wireless/wgated.tr

File name of stash password: /usr/lpp/wireless/wgated.tr

SSL Version 2 timeout: 100

SSL Version 3 timeout: 600

Figure 20-12 WAP proxy SSL settings

Installing a SSL certificate

To add the SSL certificate of an application server, you must receive the certificate either in binary or in Base64 format, and store it in a temporary directory (for example, /tmp) on the system where Connection Manager is installed:

1. Make sure you are logged in as root and start the IBM Key Management utility by entering the following command on a command line:

   ```
   wg_ikeyman
   ```

2. Open the WAP proxy key database that is listed in the SSL tab of the WAP proxy configuration and enter your password.
3. Select **Signer Certificates** from the drop-down list.
4. Click the **Add** button.
5. Select the format of the certificate from the drop-down list in which you received the certificate from the application server.
6. Enter the filename and location of the certificate in the input fields; you can also use the **Browse** button to locate the file.
7. Click **OK** and close the IBM Key Management utility.
8. Restart the Connection Manager to activate the changes.

21

Roaming

The objective of this chapter is provide information about the roaming support and how IBM WebSphere Everyplace Connection Manager (WECM) manages this service. The topics covered in this chapter include:

- Overview
- Scenarios
- Roaming in IBM WebSphere Everyplace Connection Manager
- Client configuration
- Examples
- Considerations

21.1 Overview

Roaming in its simple concept is the function that permits mobile devices the ability to move between different access points keeping the communication continuously.

This can be accomplished because this new support keeps the access points connected and they share information about the connected user. When the signal quality is better in the secondary access point, the user automatically roams from the original access point to the second one to take advantage of the better data transfer quality.

In cellular networks, roaming is the ability to move between cells, systems, or service providers. Moreover, this term is used when this ability to move occurs between service providers, and often represents that you move to an area different than the one where you are the primary subscriber.

Roaming between networks is emerging as a resource wanted by users and corporate enterprises. It reduces needed users skills to keep connections active, reduces loss time of mobile users, and therefore increases productivity and leverage of the user mobility.

In wireline data networks, it is intuitive the perception that if the user disables a communications adapter, then data communication is lost. Moreover, if the user has two adapters in the same device (for example, two Ethernet adapters) when the user changes the connection from one adapter to the other, the current communication is also lost, even if the user is keeping the connection in the second adapter.

If the user is downloading a file, the download stops. This happens because the network is not ready to understand that the user of the two adapters is the same user. In this scenario, the network gateway drops the packets sent for the IP address of the first network because this IP is no longer active.

Note: The network listens for IP addresses and not for assigned users.

When using wireless networks, the user in movement often loses the connection to a far away access point, or also simply by changing to another area covered by a different subnet. In this scenario for example, the new subnet is not ready to recognize the original IP so the session is broken.

The roaming function in data networks provides the ability for the user to move himself from one network to another without this loss of session. In an ideal situation, this should be done by moving between different subnets, from inside to outside of the intranet and back, changing the adapter, changing the network, and in some cases even changing the service provider.

21.2 How it works

Most of the networks are not configured to accept roaming because the packets are sent by the application server for a specific source IP address identified in the header. When a user connects to an application server, the application server will answer to the source IP address received in the header of the packet. Once the user's device changes the IP address, the application server is no longer able to reach the source IP address so packets are dropped by the network.

IBM WebSphere Everyplace Connection Manager works in the middle of this communication using its own headers to send the request to application servers. The IBM Mobility Client adds a header to the packets that makes the server understand that the user is still the same even when the source originates using a different IP address.

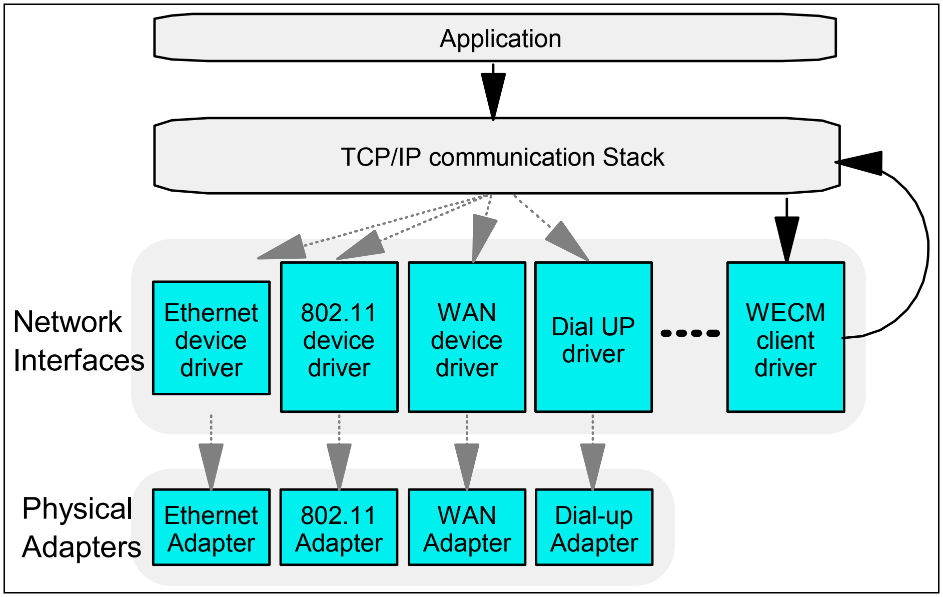


Figure 21-1 TCP/IP stack relation with drivers

In WebSphere Everyplace Connection Manager this happens because the IBM Mobility Client installs a network driver in the user's device, and this driver inserts a virtual IP header to the packets that indicate Connection Manager server as the destination. For more information refer to Chapter 11, "Mobility Client connectivity" on page 191.

With this implementation the Connection Manager server is able to manage packets and forwards them to the correct destination. In turn, responses are also sent back to the correct requesting source, even after network changes.

21.3 Scenario

This section describes a scenario where roaming can be useful. The entire scenario is focused from both perspectives: the provider and the users.

Independent user of a Internet Service Provider (ISP)

This type of user usually logs in to the ISP using some authentication process to be able to access provided services. Once the user has subscribed to these services, one usually receives the rights to access the selected services regardless of the network connectivity one is using.

In fact, some users may have connectivity to the Internet using more of one type of adapters or networks as illustrated in Figure 21-2.

The most frequently connections used by regular users are the xDSL, dialup, and WSP such as GPRS and 1xRTT. The authenticated user may be using some type of service such as watching a movie at the time that one needs to move them self to another physical location. This ability to move around in many cases may need a new network connection to keep the user continuously connected to the server at all times.

In general terms, without roaming capabilities the user will lose the connection with a network change and consequently will lose minutes of a movie. Moreover, depending on the time-out established by the ISP, the user may have considerable delays when trying to go through a fast re-authentication because the server may take extra time to disconnect the user so a new connection can be established.

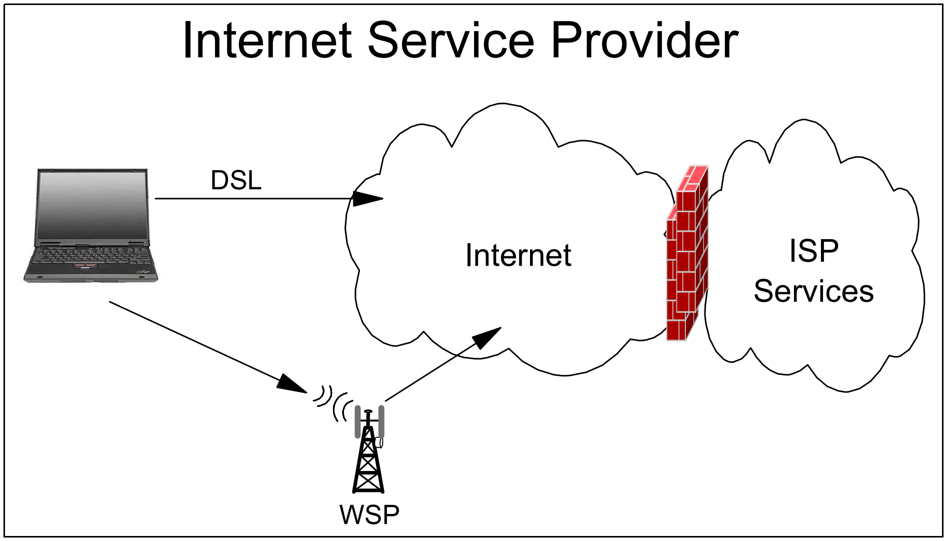


Figure 21-2 ISP scenario

Telco scenario

In specific scenarios when using Wireless Service Providers (WSPs), the ability of users to move around is obvious. In many cases, WSPs may have a structure that uses different types of networks, for example, areas covered by 2 G networks, other areas covered by 2.5 G, or 3 G networks and even PWLAN networks.

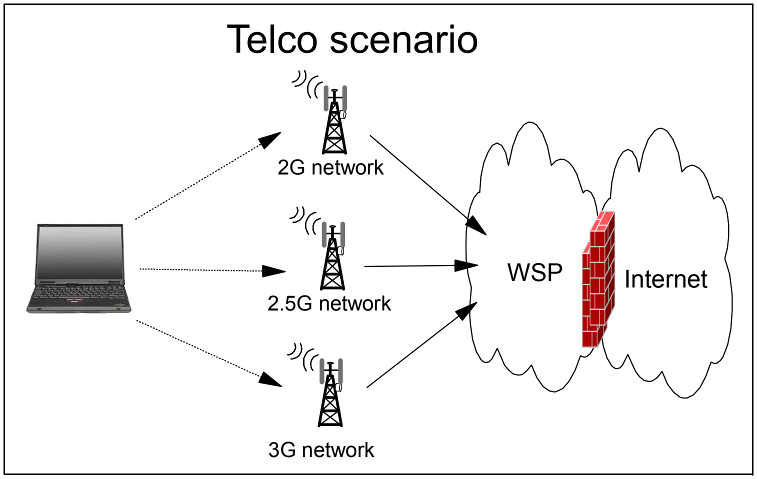


Figure 21-3 WSP scenario

In this scenario, the WSP user should be able to keep the connection for each of the networks they are using. In the general case, if the user moves around from one type of network to another, and the WSP system is not set up to accept this change of networks, the session will break. This happens because each network has its own characteristics and structure inside the WSP. This scenario is illustrated in Figure 21-3.

A simple corporate scenario

In a corporate scenario, employees often must walk away from their offices, for example to work in a different office or to go to a meeting room as illustrated in Figure 21-4. In this case, the device must be disconnected of the wired network during the path to the other office. Moreover, in the middle of the path to a new location there may be some type of wireless network such as 802.11b. Eventually, in the alternate office a new connection has to be used such as a wired connection using different network standards or even if using the same standards, the wired connection may be in a different subnet.

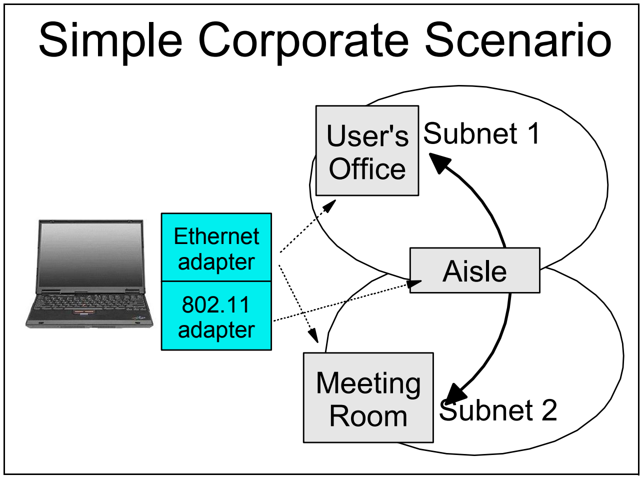


Figure 21-4 Simple Corporate Scenario

During this path, for example if the user is downloading a presentation one needs to use in the meeting room, the download process will need to be started again. Of course, this happens if the system is not prepared to roam the user from one network to another.

Complex corporate scenario

In this typical scenario, a sales executive user is for example constantly moving around from inside to outside of the company premises. In this situation, the user is constantly changing networks and many different network types can be used.

The user goes across firewalls from inside to outside and back will use the intranet network, WSP network, and even a customer network as shown in Figure 21-5.

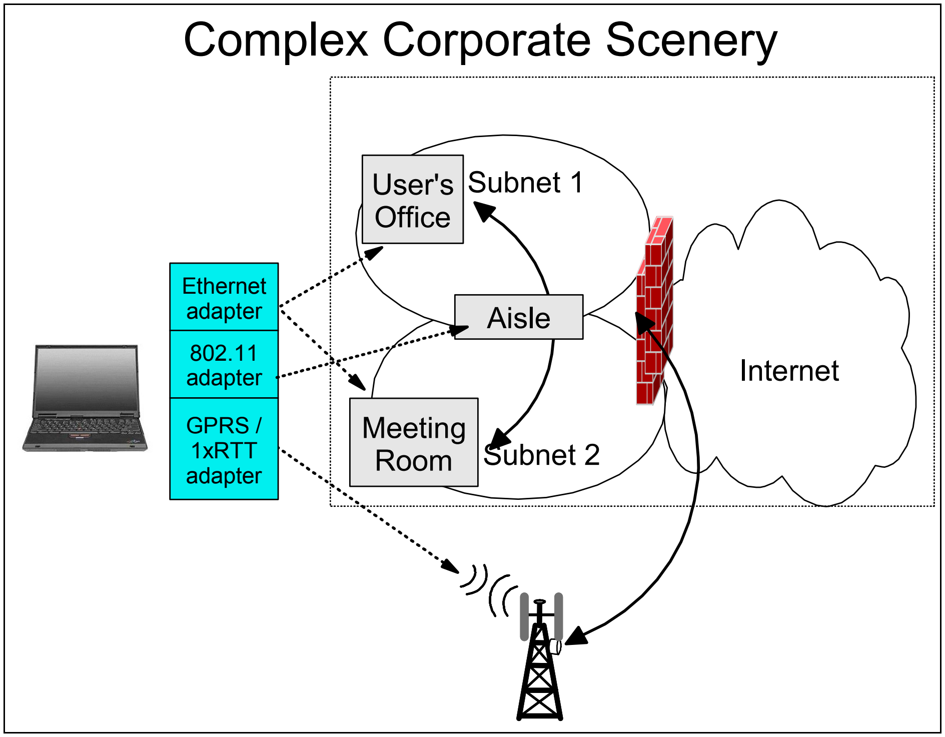


Figure 21-5 Complex corporate scenario

This situation makes the user loose the connection every time there is a network change. For example, the user may be in an important video conference with the company general manager when one needs to go to a customer meeting. Without the right network resources, the video conference will be lost.

Benefits of roaming

As described in these described scenarios, the user will loose the network connection in all cases when changing networks. In addition, this problem may also increase the amount of data traffic in the network because some jobs must be started again. This situation can also generate bad user impressions, specially with ISPs and WSP. In short, the productivity of the company can be affected for the time lost.

In a roaming environment, the ISP customer will not loose the movie, the WSP will not loose the connection in the road, and the corporate user will not loose the video conference or meeting. In short, when using roaming capabilities, the user is able to keep sessions even when the user moves from one network to another. In the long term, this will save time, money, satisfaction, and productivity.

21.4 Roaming in Everyplace Connection Manager

The IBM WebSphere Everyplace Connection Manager is a solution that provides seamless roaming to mobile devices independent of the network that the user has connectivity to. A variety number of protocols that are described in Chapter 1, “Introduction” on page 1, can be used in roaming. This solutions make the users able to roam from inside intranet to outside Internet, public network and back.

This may happen using wired or wireless networks from any service provider. These networks can be owned by service providers or corporates, and do not need any modification in the service provider network. The solution can also be installed in both service providers or enterprises, providing the seamless roaming to its users.

The user can be able to use an kind of communication protocol, the adapter of this protocol must be installed in device. Anyway, if the user has two or more adapters of the supported communication protocols, they can be used also to have seamless roaming between communication protocols.

IBM WebSphere Everyplace Access can also provide seamless roaming over enterprise firewall. It provide the ability to roam from inside intranet to Internet and back over the firewall.

21.5 Client configuration

The IBM Mobility Client is already preconfigured to this function. To install the IBM Mobility Client, refer to Chapter 10, “Mobility Clients” on page 171. Once the client installation is completed and the connection is created some setup steps may be done to improve the roaming function:

1. Right-click the icon of the desired connection and select **Properties**.

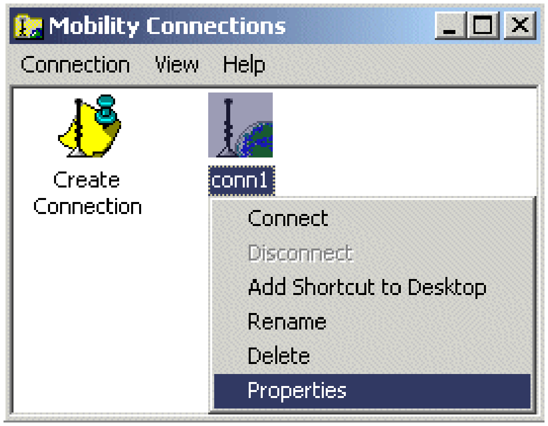


Figure 21-6 Editing preferences

2. Select the **Networks** tab.

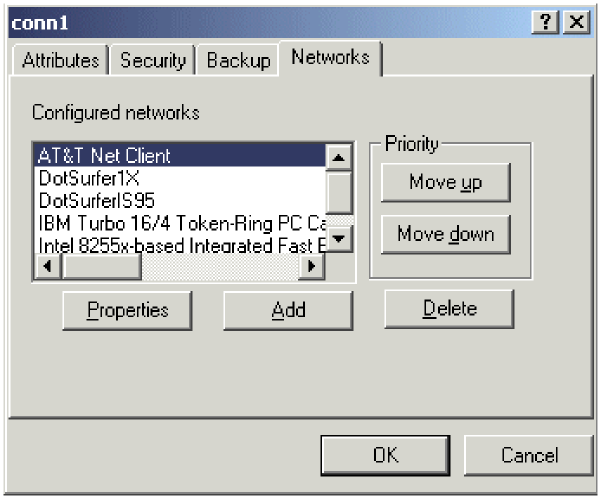
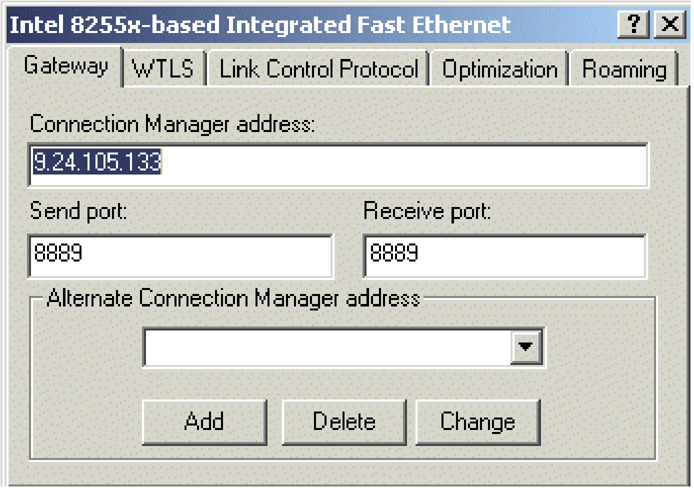


Figure 21-7 Network tab

In this tab the client will show all the network adapters selected for the connection in the installation. Once the user is in an area covered by two of the networks, the priority of the connection will respect the order described in this tab. The user can reorder the priority using the buttons **Move up** and **Move down**. This gives the user ability to chose, for example, a cheaper network or a faster network as preferred. In this tab the user can also add or delete adapters:

3. Click the **Properties** button.



Intel 8255x-based Integrated Fast Ethernet

Connection Manager address: 9.24.105.133

Send port: 8889

Receive port: 8889

Alternate Connection Manager address

Figure 21-8 Properties view

This function is used in a roaming view just as if the user intended to roam from intranet to Internet or back. This is needed because depending on the configuration of the network, the gateway may need two different IP addresses to be reached. Usually, the network in a Connection Manager environment uses architecture as in Figure 21-9.

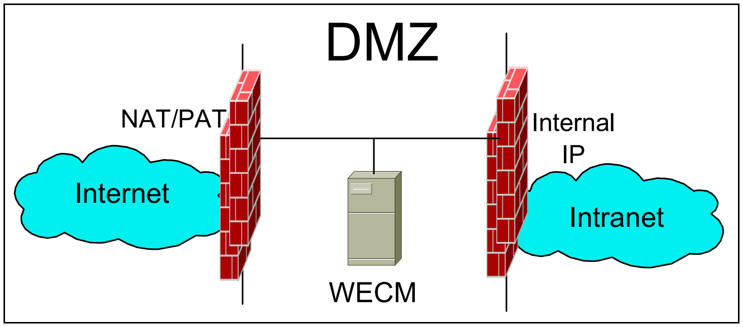


Figure 21-9 IP addresses view

When the has user access is from external networks such as the Internet, the IP used to reach the server is a Network Address Translation (NAT) or Port Address Translation (PAT) of the server IP. When the user's access is from intranet, the IP used to reach the server is the internal IP address. In fact, the connection is received by the same IP of the server, but when using NAT, the public IP forwards the packets to the private IP.

To configure this alternate server address:

1. Click the **Add** button.

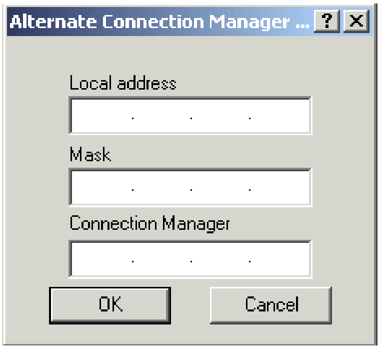


Figure 21-10 Alternate Connection Manager address

2. In Local address field, insert the IP range of the network used to connect to the gateway. For example, if with the alternate Connection Manager address the user wants to use is the internal one, and the internal IP of the user is 192.168.0.x, the local address field can be 192.168.0.x. If the IP is designed by DHCP and the user has one different IP each connection, then the range can be used as 192,168.0.0.
3. In mask the field one can use the required mask to the fixed IP or a mask that defines the range in case of DHCP.
4. In the Connection Manager field, insert the desired alternate IP address, and the internal IP address.
5. Click **OK**.

Just as a warning, the easier configuration is the one where you use the alternate Connection Manager as the internal IP. This is because the IP ranges are controlled and known. By the way, the external IP addresses are not known previously, so the IP used for NAT is better used as the default Connection Manager IP in the Connection Manager address field shown in Figure 21-8.

6. Select the **Roaming** tab and determine the preferences for each field. The meanings of the fields are:
   a. **Time to wait field**: Specifies the number of seconds that the Mobility Client waits after the network is available before the Mobility Client will roam to it and make it the active connection. If the connection is being established on the periphery of a network coverage area, this setting allows a period of time for the Mobility Client to make that the connection status does not go out of range or unavailable before it attempts to roam to it.

b. If status is Active field specifies the number of seconds that the Mobility Client connection to a network has been in Active status, and acts as a threshold value for determining which Wait interval to use when roaming from this connection to a lower priority network, if the status changes to `Out of range` or `Unavailable`. This setting helps determine if you are moving into or out of the coverage area of a network. If you are moving into coverage area, you may want to delay roaming from the network quickly because there can be a period of time when the status flips back and forth between active and inactive. This delay gives the network a chance to stabilize and establish the actual status. However, if you are moving out of a coverage area, and it indicates that the network status has been active for a given period of time and then becomes inactive, then you may want to roam from that network more quickly.

c. The Wait field specifies the number of seconds that the Mobility Client waits before roaming from this connection to another available network of a lower priority. When the network status has been active for more than the amount of time specified in the `If status is Active longer than` field and then becomes `Unavailable`, this setting determines the number of seconds the Mobility Client waits until it will roam from this network to another of lower priority. The value of this setting should be less than the value of the `Otherwise wait` field.

d. The `Otherwise wait` field specifies the number of seconds that the Mobility Client waits before roaming from this connection to another available network of a lower priority. When the network status has been active for less than the amount of time specified in the `If status is Active longer than` field, then the connection may be in the periphery of a network coverage area, and may need additional time to establish its actual status. This setting determines the number of seconds the Mobility Client waits until it will roam from this network to another of lower priority. The value of this setting should be more than the value of the Wait field.

7. Once you finish the last step, click **OK**.
8. Click **OK**.

Configuring for automatic or manual roaming

The user has the ability to set if it wants automatic or manual roaming. If the roaming is set to automatic, it will respect the priority sequence showed in 2., “Select the Networks tab.” on page 465. Else, if the roaming is set to manual, the user must change manually the desired network. By default the automatic roaming is enabled. To set it, establish connection and then:

1. Find the IBM Mobility Client connected icon near the Microsoft Windows clock.

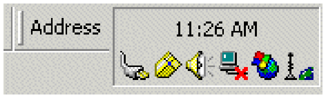


Figure 21-11 IBM Mobility Client icon view

2. Right-click the **clock**.

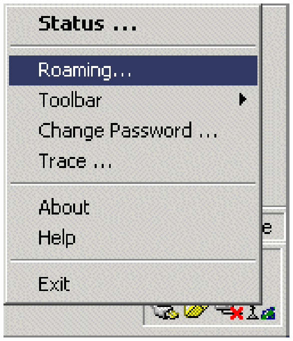
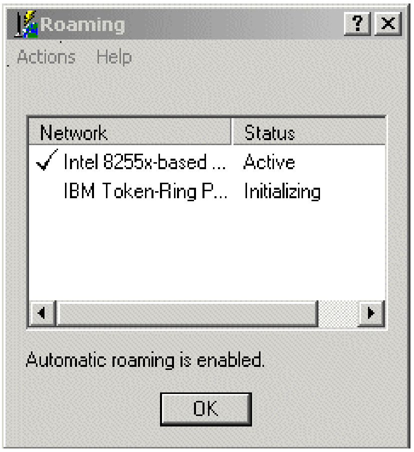


Figure 21-12 Menu view

3. Select the option **Roaming**.

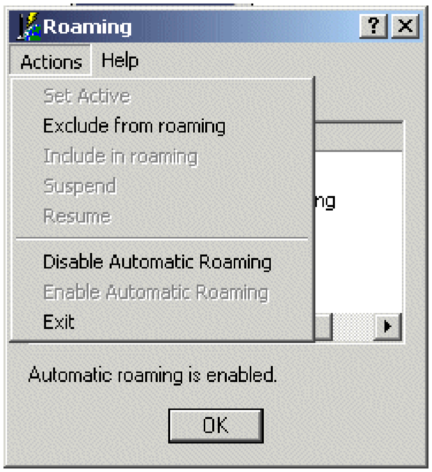


Network	Status
✓ Intel 8255x-based ...	Active
IBM Token-Ring P...	Initializing

Automatic roaming is enabled.

Figure 21-13 Roaming option view

4. Select **Actions**.



Automatic roaming is enabled.

Figure 21-14 Actions

5. Select **Disable Automatic Roaming** or **Enable Automatic Roaming**.

Roaming configuration

The server side just needs the configuration of the MNCs that can be used to access itself. If the user has a network protocol that has not an MNC configuration, the user will not be able to connect to the server even directly. By the way, if the user can connect to the server using all of the network protocols, the roaming is by default accepted. Information about the MNC configuration can be found in Chapter 8, “Administration” on page 85.

21.6 Examples

By using the same scenarios described in 21.3, “Scenario” on page 460 it is easy to understand the results provided by IBM WebSphere Everyplace Connection Manager.

In “Independent user of a Internet Service Provider (ISP)” on page 460 the user is connected to xDSL using an Ethernet adapter watching a movie. This Ethernet adapter has a real IP address linked to a virtual IP address provided by IBM WebSphere Everyplace Connection Manager. When the user disconnects the xDSL and connects using a GPRS/1xRTT adapter, the IBM Mobility Client keeps sending the header with the same virtual IP even with real/physical IP changes. This makes the Connection Manager server identify the user and knows that the user is watching a movie. With this, the user is able to continue to see the movie.

In the “Telco scenario” on page 461 the user that roams from one technology to another technology has the connection broken. But the session keeps alive for some sufficient time to make the client reconnect to the Connection Manager server. In this process, the server identifies that the client is still the same because the virtual IP is the same, so the server rebuilds the session.

In “A simple corporate scenario” on page 462 the employee must attend to a meeting and needs some more minutes to finish the presentation download. When the user disconnect from the wired network, the WLAN adapter is still connected but the application server do not know that the adapter is own by the same user. Anyway, the application server is sending the packets to Connection Manager server. As soon the Connection Manager server receive a new packet sent by the client through the new adapter, the server makes the relation of the virtual IP with the new real IP. When the user connects to a wired connection again, even being in another subnet and the adapter receiving a new IP address, the server still identifying the user. Automatically the IBM Mobility Client roams to the new network connection, respecting the priorities defined by the user.

In “Complex corporate scenario” on page 462 the user is moving from inside to outside intranet. At this moment, the Ethernet adapter connected to the intranet lose its connection and the GPRS/1xRTT adapter is the one available to communicate. In this case, the IBM Mobility Client switches the communication to the adapter available. Considering that this adapter has a different IP subnet and this IP is not internal, the client knows that the Connection Manager server IP cannot be reached using its internal IP. The client changes automatically the Connection Manager server address and this start to receive the packets by an external address redirected by NAT. At the moment that the packets reach the server, the server identifies the virtual IP, and links it to the user, and then keeps the session alive.

These examples are the most simple and real examples that prove the utility of the roaming feature. The user in an ISP can now watch the movie. The WSP can provide continuously connection to its customer. The employee can attend to the meeting on time and with all material needed to presentation. And the sales executive can finish the video conference with the company general manager in the taxi on the way to close a big deal.

21.7 Considerations

The Mobility Client will automatically roam from a higher priority network to a lower priority network when the higher priority network becomes unavailable. Unavailable means for example that a network adapter has been removed, a cable has been disconnected, or in the case of some wireless networks the user has moved out of range of the network. For IP based networks, cable disconnect and 802.11 range events require the underlying operating system support a feature known as media sense. However, you should consider the following cases:

- Windows ME, Windows 2000 and Windows XP support media sense.
- Media sense is not available on Windows 98, Windows CE HPC2000, and Pocket PC 2000 and Pocket PC 2002.
- Additionally, media sense events are not used in wireless WAN networks such as GPRS and 1xRTT on some versions of the Windows operating system. It is recommended that you verify this support before planning for seamless roaming capabilities when using these networks. That is, wireless WAN networks do not support media sense. Media sense is only supported by LAN networks such as Ethernet and WiFi.
- If the Wireless WAN NDIS WAN driver for the WAN network is written such that the they drop the circuit switched connection when out of range, then the Mobility Client will detect that the connection is down and automatically switch to a lower priority network. However, not all circuit switched wireless WAN connections behave this way. Many will stay *connected* even when out of range. So, it is really network and network-card specific and requires each network and network-card combination to be evaluated to understand how it will behave.

> **Note:** For the cases where media sense events are not available, a third party developer can extend the Mobility Client to have knowledge of network connectivity events for a specific radio modem by developing a custom DLL designed for that modem. Many radio modem vendors supply a toolkit to allow users to programatically obtain information such as signal strength and range events. By developing a status DLL that retrieves this information and passes it to the Mobility Client, the Mobility Client can be aware when a given network is no longer available.

22

Clustering

Clustering is a function provided by Connection Manager to distribute the workload originated from pervasive devices to access application servers and vice-versa. This functionality allows Connection Manager scalability.

This chapter describes the following topics:

- Overview
- Sample scenario
- Connection Manager configuration for principal and subordinate nodes
- Mobility Client configuration
- Running the sample scenario
- Troubleshooting

22.1 Overview

Clustering enhances the wireless networks responsiveness. It also offers a more robust design against failures. It is usually used in large enterprises having a significant quantity of pervasive devices, and therefore with very high traffic. In a Connection Manager environment, this function is usually implemented with one principal node, and several subordinate nodes as shown in Figure 22-1.

- Principal node

  The principal node is responsible for the connection between pervasive devices and the subordinate nodes. It runs only the Mobile Network Connections (MNCs) resources. It does not process the packets, but only dispatches the traffic to the subordinate nodes.

- Subordinate node

  The subordinate nodes process packets received from the principal node, delivering them to the application servers, and vice-versa. They run the Mobile Network Interfaces (MNIs) resources, and can be grouped in a *cluster group* resource. Dynamic scalability is provided by adding subordinate nodes to the group on the fly.

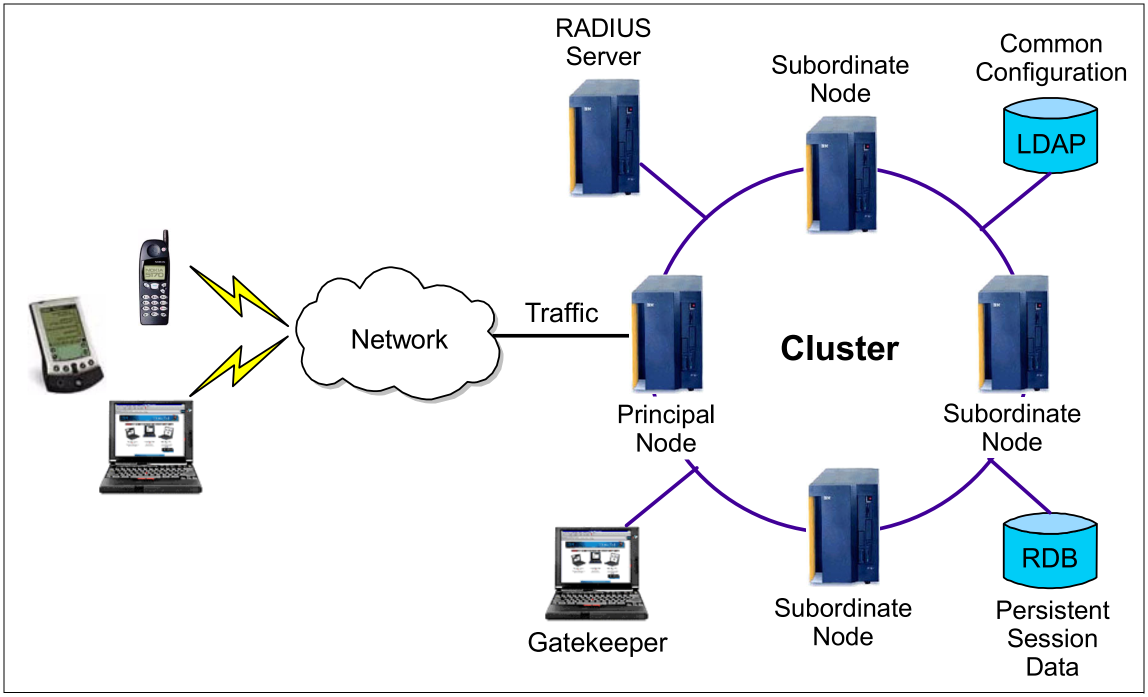


Figure 22-1 Cluster architecture

Note: In order to reach high availability, it is necessary to duplicate the principal node in an HACMP configuration, and mirror the LDAP and database servers. For information about High Availability Cluster Multi-Processing for AIX (HACMP) see Chapter 23, “Implementing HACMP with Connection Manager” on page 503.

In Connection Manager, clustering is used for Mobility Client traffic and WAP client traffic. For WAP devices, this scheme can offer a more extended secure connection capability when using WTLS. You should also notice that in this type of implementation there is only one principal node per cluster manager. In addition, when a principal node shuts down, the cluster manager notifies subordinate nodes that they should abort any pending transactions that get routed back through the principal node.

22.1.1 Subordinate node states

Subordinate nodes can be configured to be in one of the following three states:

- **Active**

  The normal mode of operation in which the cluster manager dispatches traffic according to the distribution algorithm configured for the subordinate node.

- **Defined**

  A mode of operation in which the cluster manager immediately notifies the subordinate node that it should abort any pending transactions.

- **Maintenance**

  A mode of operation in which the cluster manager does not route new traffic to this subordinate node, allowing the traffic to terminate after all pending transactions are completed. To verify that all traffic is terminated, use the wg_monitor utility and check the number of active sessions.

In addition, subordinate nodes can be pooled into cluster groups and principal nodes can be configured to dispatch traffic only to subordinate nodes in these groups. By default, if a cluster manager has no groups to manage, it manages all subordinate nodes that are not assigned to a cluster group.

22.1.2 Dispatching

A principal node dispatches traffic on connection basis to subordinate nodes. At connection initialization, when the user needs to be authenticated, if a subordinate node fails to resolve user lookup, the dispatcher does a new authentication attempt to the next available subordinate node.

Note: To dispatch traffic for a messaging MNC, make sure that the MNC is installed and configured on the same system as the messaging services.

A principal node initiates communication to subordinate nodes. As part of the role of receiving and dispatching traffic, a principal node maintains two-way communication with subordinate nodes.

A control connection is established between the principal node and each subordinate node. It may be a clear TCP or a secure connection using Secure Sockets Layer (SSL). This connection allows the principal node to search for subordinate nodes heartbeat and workload. It causes the principal node to choose which subordinate node to use for that connection. The principal node may also dispatch the traffic according to the Mobile Network Connection (MNC), or the Device Identifier. This control connection may be protected with SSL (on subordinate nodes select **TCP/SSL** as the internode transport protocol; the key databases on principal and subordinate nodes are cm.trusted.kdb).

A Cluster Management Protocol (CMP) header is added to the data packet to pass some connection data from the MNC to the subordinate node.

A subordinate node may be set in maintenance mode. No more new connections are dispatched to this node, and pending connections complete normally.

Note: You may use the **`wg_monitor`** utility to check the number of active connections before disabling completely the subordinate node.

Distribution algorithms

Subordinate nodes send load information to principal nodes and control when to begin or end accepting traffic from principal nodes based on a configured distribution algorithm.

The distribution algorithms include:

- Round-robin

  The principal node continuously repeats the sequence of distributing traffic to a series of subordinate nodes, one after the other.

- Weighted round-robin

  The principal node continuously repeats the sequence of distributing traffic to a series of subordinate nodes, based on configured CPU utilization thresholds called low and high water marks.

- Device/MNC based

  The principal node distributes traffic to subordinate nodes based on the MNC or unique device identifier from which it came.

For configuration details about distribution algorithms see Figure 22-15 on page 486, and Figure 22-16 on page 487.

22.2 Sample scenario

The sample scenario described in this chapter has a principal node and two subordinate nodes. It can also be extended with a minor effort to run in a multiple subordinate node scenario.

In this sample scenario, the following machines are used as shown in Figure 22-2:

1. Principal node
   - IBM AIX version V5.2.0.0 with Maintenance Package 01
   - IBM DB2 Universal Database 8.1 Fixpak 5 (client version)
   - IBM Directory Services V5.1 (client version)
   - IBM WebSphere Connection Manager V5.0.1.1
2. Subordinate node 1
   - IBM AIX version V5.2.0.0 with Maintenance Package 01
   - IBM DB2 Universal Database 8.1 with Fixpak 5 (client version)
   - IBM Directory Services V5.1 (client version)
   - IBM WebSphere Connection Manager V5.0.1.1
3. Subordinate node 2
   - IBM AIX version V5.2.0.0 with Maintenance Package 01
   - IBM DB2 Universal Database 8.1 with Fixpak 5 (client version)
   - IBM Directory Services V5.1 (client version)
   - IBM WebSphere Connection Manager V5.0.1.1
4. LDAP and database server
   - Microsoft Windows 2000 Advanced Server ServicePack 4
   - IBM DB2 Universal Database 8.1 Fixpak 5 (server)
   - IBM Directory Services V5.1 (server)
5. Application Server 1
   - Microsoft Windows 2000 Advanced Server ServicePack 4
   - IBM HTTP Server
   - FTP server
6. Gatekeeper
   - Microsoft Windows 2000 Professional ServicePack 4
   - IBM Connection Manager Gatekeeper V5.0.1.1
7. Mobility Client

- Microsoft Windows XP Professional
- IBM Mobility Client V5.0.1.1

The sample scenario is illustrated in Figure 22-2. Notice the following considerations:

- Only one application server is used, and therefore the route configured in the subordinate node MNI is for a single host only (mask is 255.255.255.255). For multiple servers you should configure a route for a subnet.
- For simplicity, the configuration is shown for one subordinate node only, other subordinate nodes will have similar configuration.
- The IP address range for the Mobility Clients is configured in the subordinate node MNI.
- The routes for the Mobility Clients are configured in the subordinate node MNI.
- LDAP directory is shared between the principal node and all subordinate nodes.
- Subordinate node (m10df5cf) and the application server machine are in the same subnet. For details and requirements, see 22.4, “Application server requirements” on page 492.

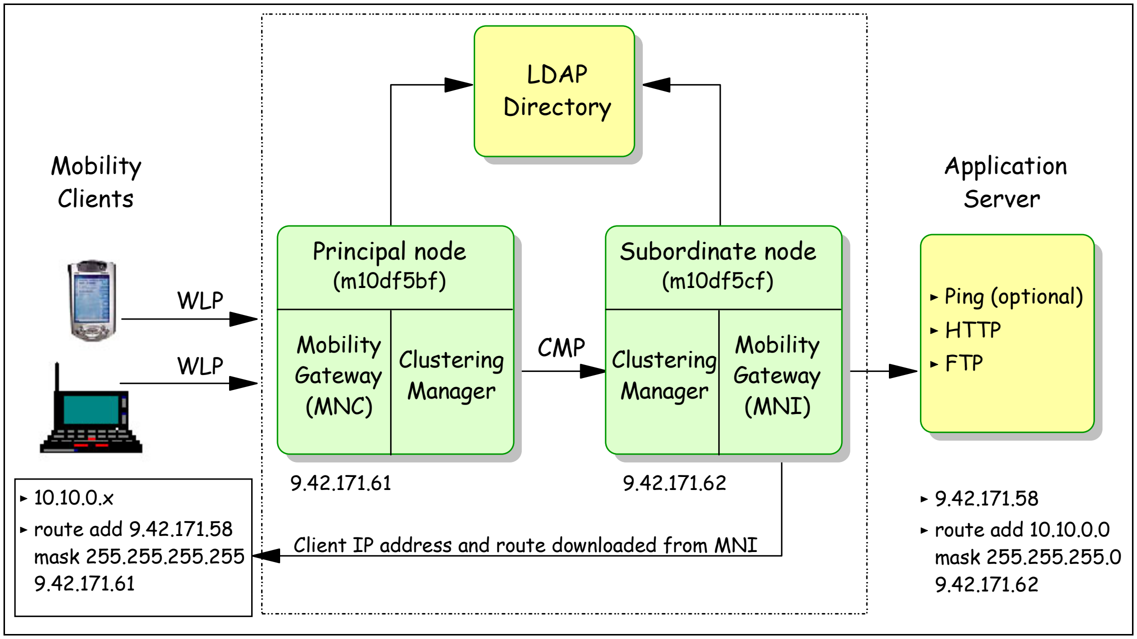


Figure 22-2 Sample scenario

In this scenario all the Connection Manager resources, and principal and subordinate nodes are defined remotely on the same LDAP server. In addition, in this sample scenario users are also remotely located in the same LDAP server.

22.3 Connection Manager configuration

Principal and subordinate nodes are configured using the Gatekeeper. This sample scenario assumes that Connection Manager resources have already been created. If it is not the case, create them and point the DSS configuration to the remote LDAP server.

22.3.1 Principal node configuration

In this section, Connection Manager is configured as a principal node in a clustering environment. The configuration includes a required MNC. For example:

1. Right-click the principal node resource and select **Properties.**

   Note: The principal node name in this sample scenario is m10df5bf.

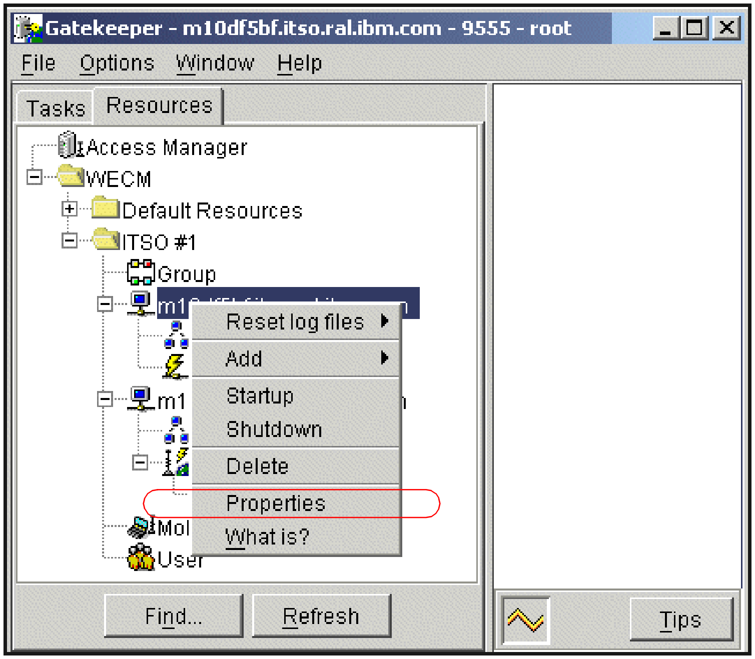
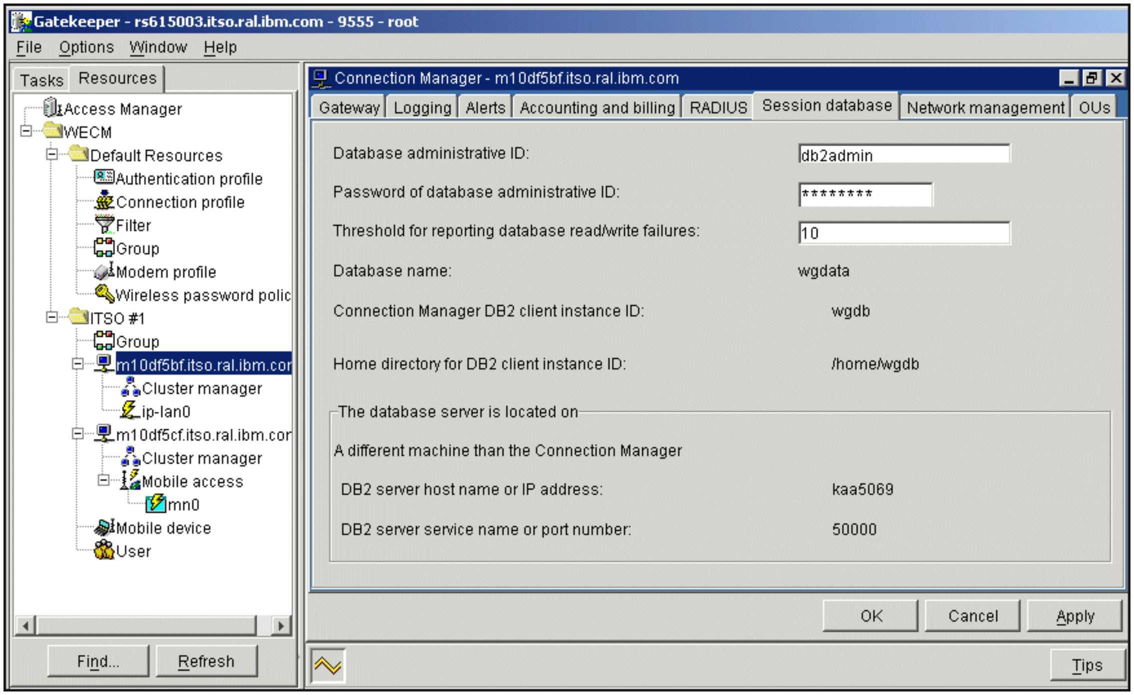


Figure 22-3 Select principal node

2. Select the session database tab and make sure Connection Manager is pointing to the correct database, and that the user ID and password are set according to the information provided by the database administrator. Connection Manager uses this database for persistent data.



Database administrative ID: db2admin

Password of database administrative ID: ********

Threshold for reporting database read/write failures: 10

Database name: wgdata

Connection Manager DB2 client instance ID: wgdb

Home directory for DB2 client instance ID: /home/wgdb

The database server is located on

A different machine than the Connection Manager

DB2 server host name or IP address: kaa5069

DB2 server service name or port number: 50000

Figure 22-4 Session database configuration

Cluster Manager configuration

In this section Cluster Manager is configured to define this machine as a principal node. The subordinate nodes area also selected.

For example, execute the following steps:

1. Right-click the principal node Cluster Manager resource, and select **Properties** as illustrated in Figure 22-5.

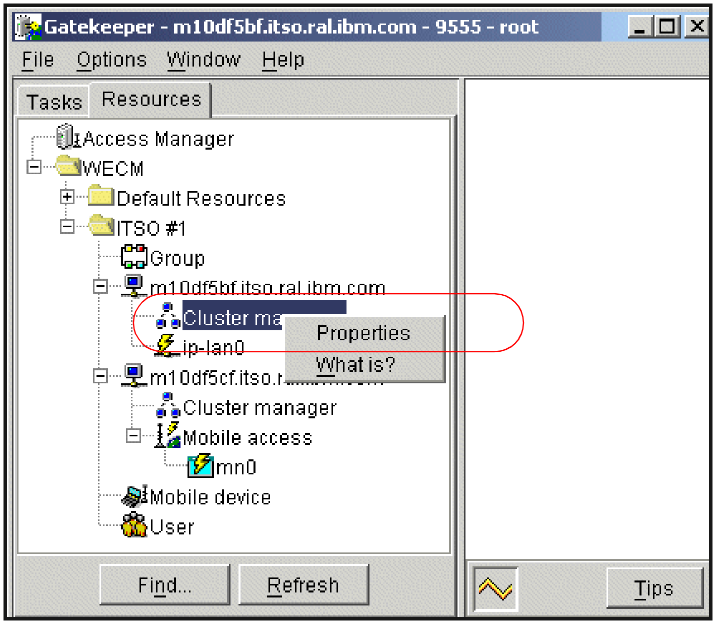
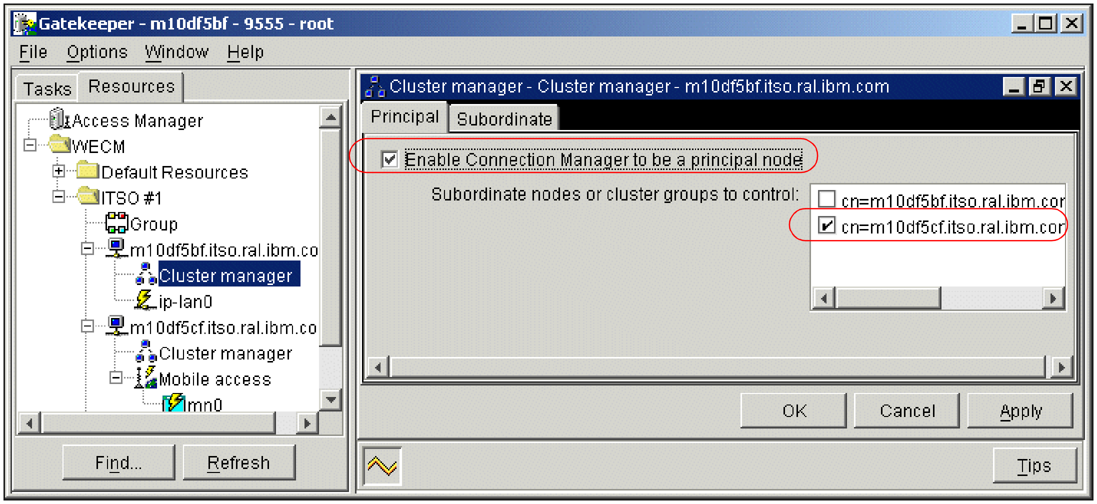


Figure 22-5 Select Cluster Manager properties in principal node

2. Select the **Principal** tab, check the **Enable Connection Manager to be a principal** node. Select also the Connection Manager machines you want to be subordinate nodes of this principal node as shown in Figure 22-6.



☑ Enable Connection Manager to be a principal node

Subordinate nodes or cluster groups to control:

☐ cn=m10df5bf.itso.ral.ibm.cor

☑ cn=m10df5cf.itso.ral.ibm.cor

Figure 22-6 Setup Connection Manager as a principal node

3. Since this is a principal node, for verification click the Subordinate tab and make sure the Enable Connection Manager option is not selected.

MNC configuration

In this section a Mobile Network Connection (MNC) is configured as required by the principal node. For example perform the following steps:

1. Right-click on the **principal node**, and add a Mobile Network (MNC) as shown in Figure 22-7.

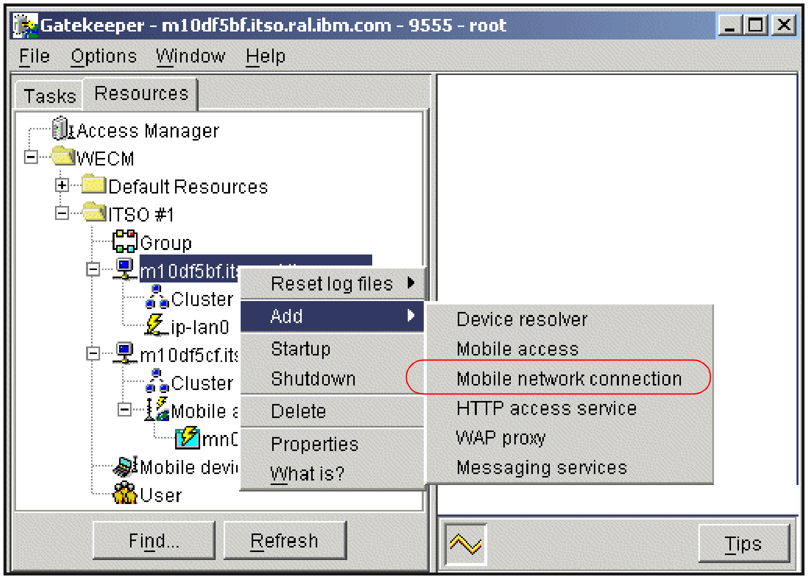
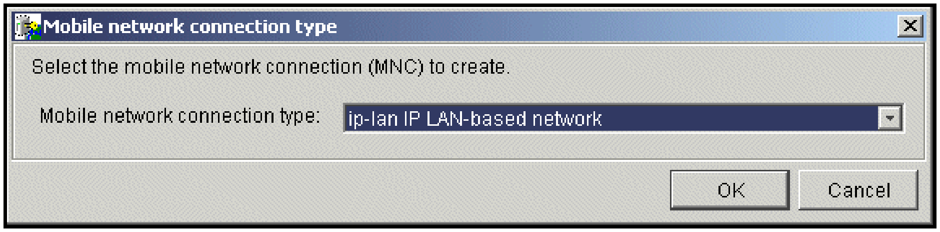


Figure 22-7 Add a Mobile Network Connection (MNC)

2. When adding the MNC you will be required to enter the connection type. For this sample scenario, the IP LAN based connection is selected as the connection type as shown in Figure 22-8.

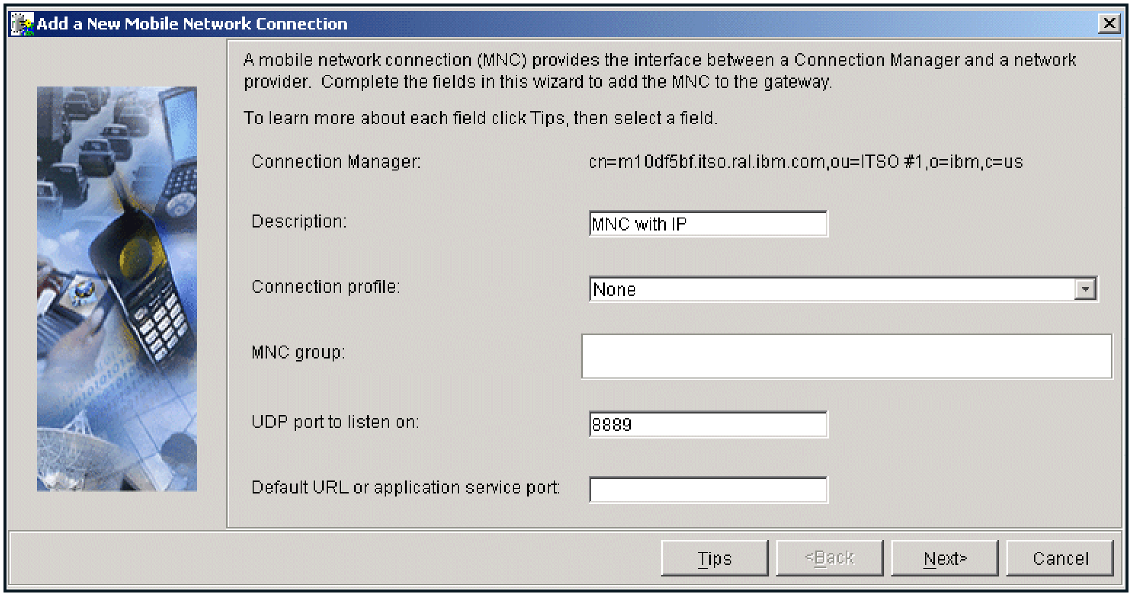


Select the mobile network connection (MNC) to create.

Mobile network connection type: ip-lan IP LAN-based network

Figure 22-8 Selecting the Mobile Network Connection type

3. The MNC configuration for this scenario is illustrated in Figure 22-9.


Add a New Mobile Network Connection

A mobile network connection (MNC) provides the interface between a Connection Manager and a network provider. Complete the fields in this wizard to add the MNC to the gateway.

To learn more about each field click Tips, then select a field.

Connection Manager: cn=m10df5bf.itso.ral.ibm.com,ou=ITSO #1,o=ibm,c=us

Description: MNC with IP

Connection profile: None

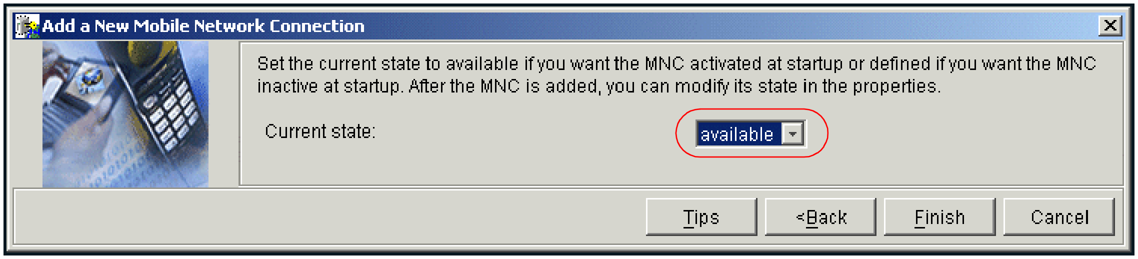
MNC group:

UDP port to listen on: 8889

Default URL or application service port:

Figure 22-9 Mobile Network Connection configuration in principal node

4. Make sure you set the MNC current state to `available` if you want the MNC activated at startup time.


Add a New Mobile Network Connection

Set the current state to available if you want the MNC activated at startup or defined if you want the MNC inactive at startup. After the MNC is added, you can modify its state in the properties.

Current state: available

Figure 22-10 MNC available option

22.3.2 Subordinate node configuration

In this section we provide a sample subordinate node configuration for this scenario. For example:

1. As illustrated in Figure 22-11, right-click the subordinate node Connection Manager resource and select **Properties**.

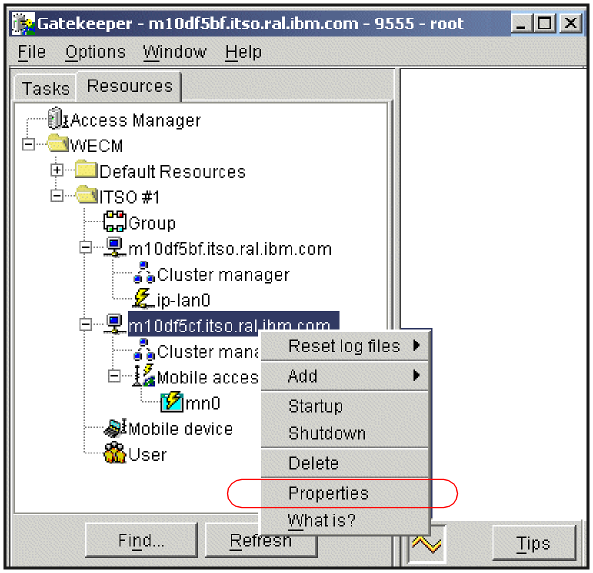
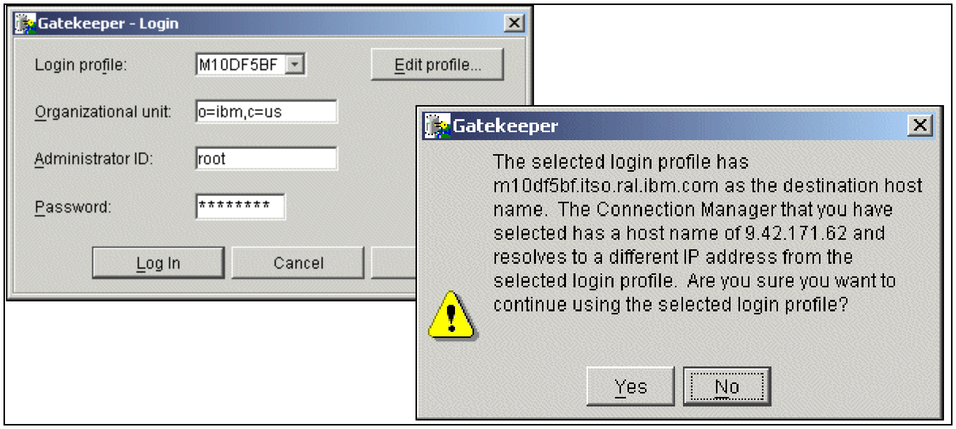


Figure 22-11 Select subordinate node

2. Log in and click **Yes** to the information message as shown in Figure 22-12. For this scenario the subordinate node address is 9.42.171.62.



Login profile: M10DF5BF

Organizational unit: o=ibm,c=us

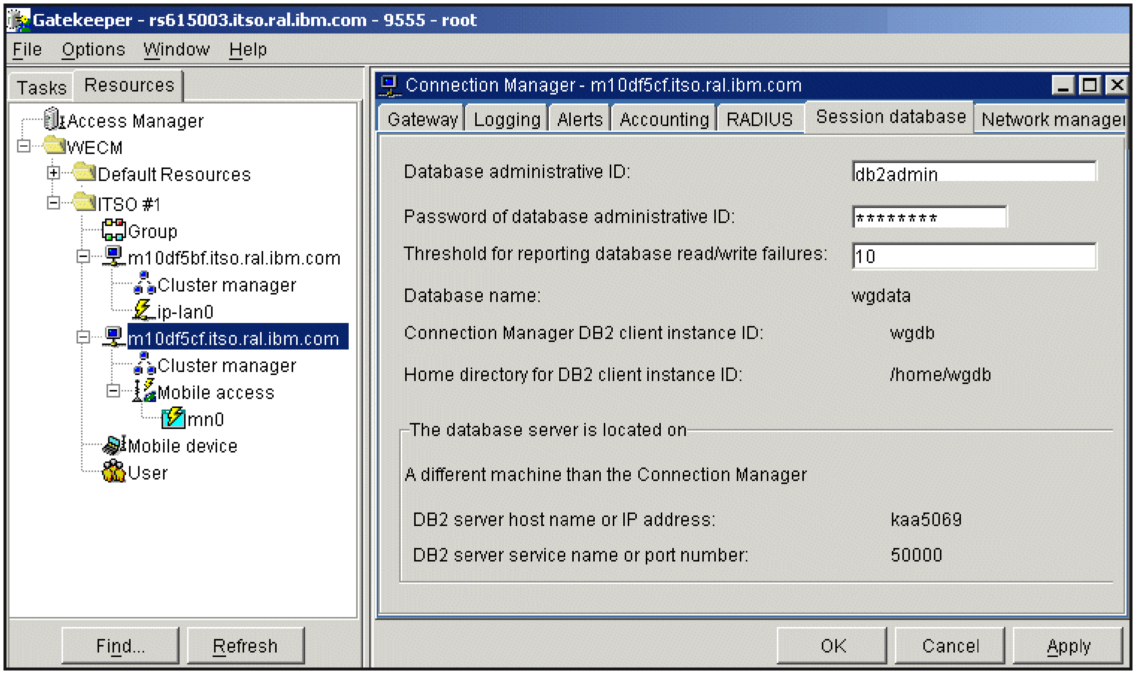
Administrator ID: root

Password: ********

The selected login profile has m10df5bf.itso.ral.ibm.com as the destination host name. The Connection Manager that you have selected has a host name of 9.42.171.62 and resolves to a different IP address from the selected login profile. Are you sure you want to continue using the selected login profile?

Figure 22-12 Subordinate node login (1 of 2)

3. Select the **Session Database** tab in the subordinate node and make sure Connection Manager points to the correct database used for persistent data. In this scenario it is a remote database and the user ID and password should be the same as provided by your database administrator.


Gatekeeper - rs615003.itso.ral.ibm.com - 9555 - root
File Options Window Help
Tasks Resources
Access Manager
WECM
Default Resources
ITSO #1
Group
m10df5bf.itso.ral.ibm.com
Cluster manager
ip-lan0
m10df5cf.itso.ral.ibm.com
Cluster manager
Mobile access
mn0
Mobile device
User
Find...
Refresh
Connection Manager - m10df5cf.itso.ral.ibm.com
Gateway Logging Alerts Accounting RADIUS Session database Network manager
OK Cancel Apply

Database administrative ID: db2admin

Password of database administrative ID: ********

Threshold for reporting database read/write failures: 10

Database name: wgdata

Connection Manager DB2 client instance ID: wgdb

Home directory for DB2 client instance ID: /home/wgdb

The database server is located on

A different machine than the Connection Manager

DB2 server host name or IP address: kaa5069

DB2 server service name or port number: 50000

Figure 22-13 Session database configuration in subordinate node

4. Right-click the subordinate node cluster manager resource, and select **Properties**.

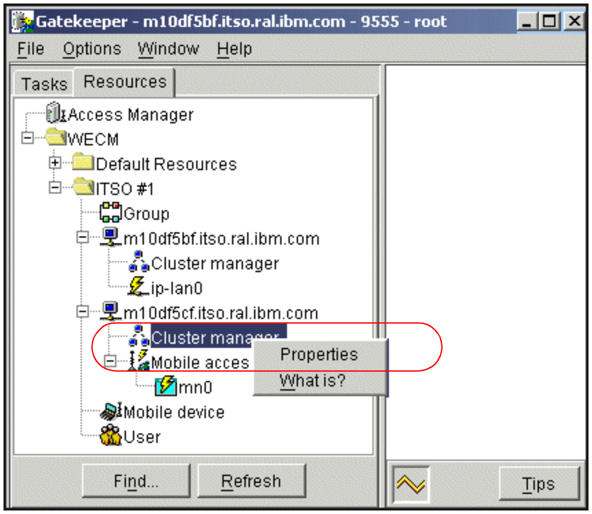
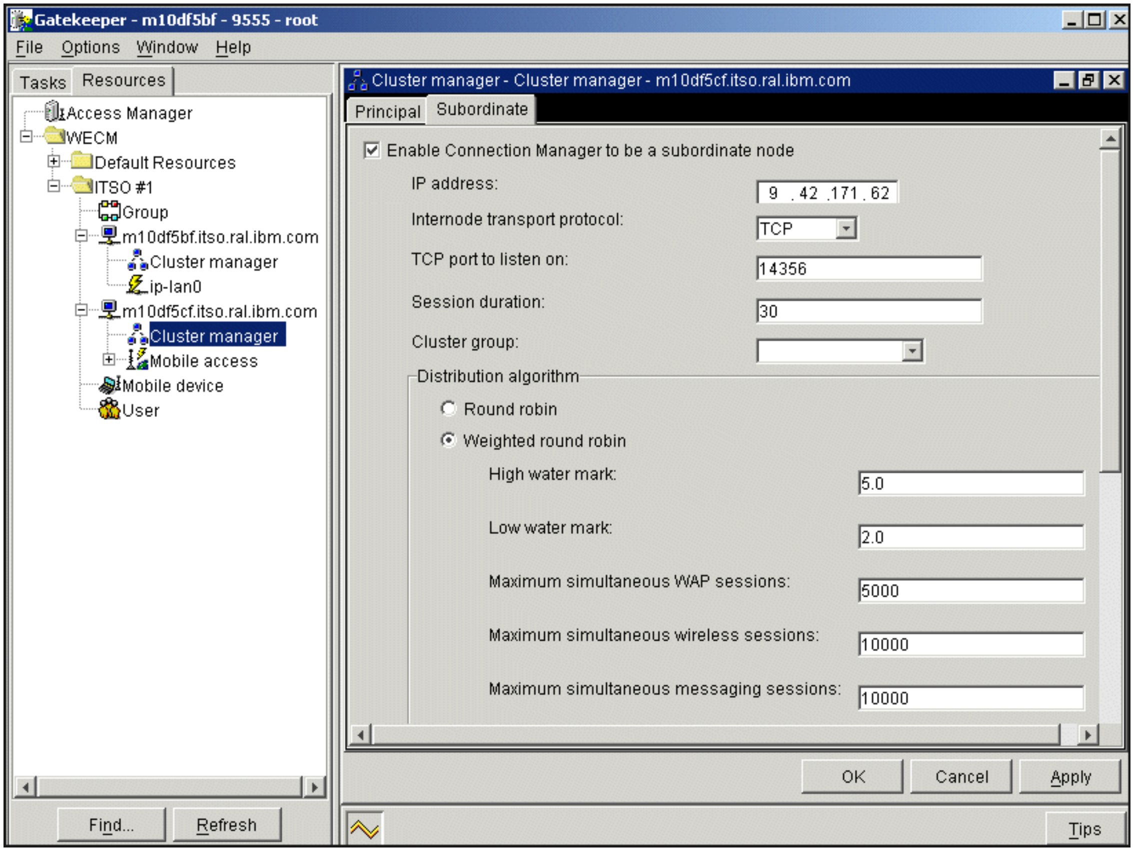


Figure 22-14 Select Cluster Manager in subordinate node

5. Click the **Subordinate** tab and select **Enable Connection Manager to be a subordinate node**. Enter the subordinate node configuration or take default values for the subordinate node IP address, transport protocol, port number, session duration, cluster groups if using groups, and the distribution algorithm. The configuration values for the subordinate node in this sample scenario are shown in Figure 22-25.



☑ Enable Connection Manager to be a subordinate node

IP address: 9 . 42 . 171 . 62

Internode transport protocol: TCP

TCP port to listen on: 14356

Session duration: 30

Cluster group:

Distribution algorithm

○ Round robin

● Weighted round robin

High water mark: 5.0

Low water mark: 2.0

Maximum simultaneous WAP sessions: 5000

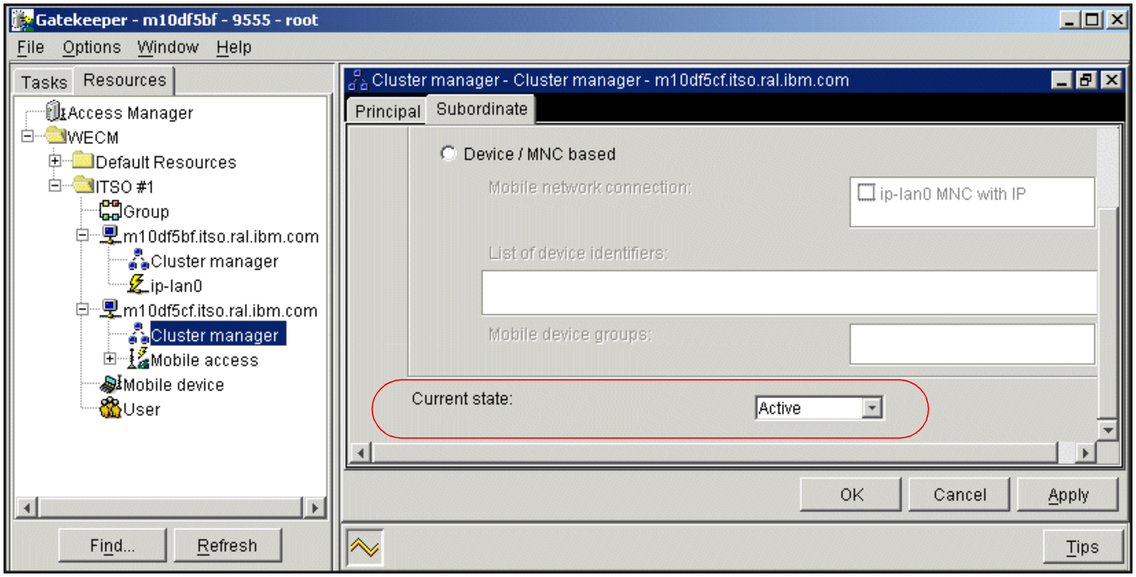
Maximum simultaneous wireless sessions: 10000

Maximum simultaneous messaging sessions: 10000

Figure 22-15 Setup Connection Manager as a subordinate node (1 of 2)

Note: Since this is a subordinate node, make sure the Enable Connection Manager to be a principal node option in the Principal tab is not checked.

6. Make sure the subordinate node is configured to be active at startup. This is the default value as shown in the second part of subordinate node configuration illustrated in Figure 22-16.



○ Device / MNC based

Mobile network connection: □ ip-lan0 MNC with IP

List of device identifiers:

Mobile device groups:

Current state: Active

Figure 22-16 Setup Connection Manager as a subordinate node (2 of 2)

Adding the Mobile Network Interface (MNI)

In this section we describe a Mobile Network Interface (MNI) configuration required to support Mobility Clients in the subordinate node. The MNI will define a contiguous range of IP addresses for the supported Mobility Clients connected to the principal node.

For example:

1. Right-click the **subordinate node Connection Manager** and select the option **Add -> Mobile access** resource as shown in Figure 22-17.

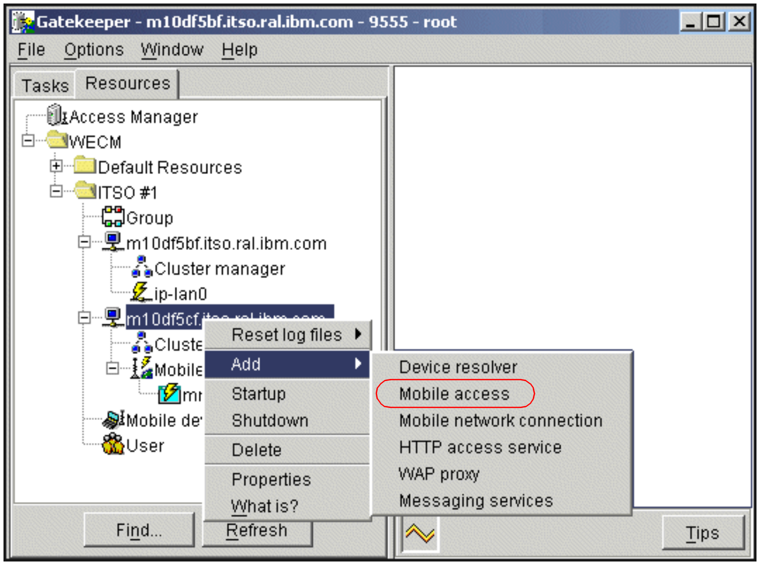
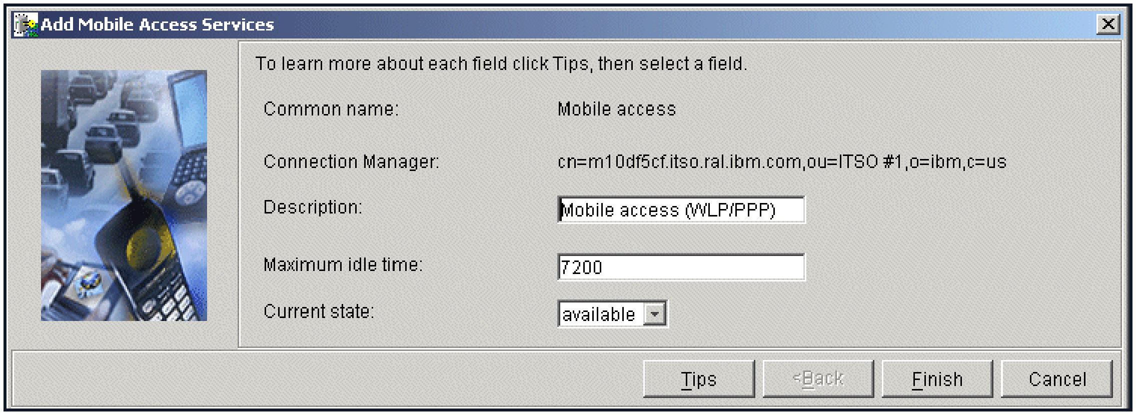


Figure 22-17 Select option to add a mobile access resource

2. Provide a description, maximum idle time, and current state. For this scenario default values are used. Click **Finish**.



To learn more about each field click Tips, then select a field.

Common name: Mobile access

Connection Manager: cn=m10df5cf.itso.ral.ibm.com,ou=ITSO #1,o=ibm,c=us

Description: Mobile access (WLP/PPP)

Maximum idle time: 7200

Current state: available

Figure 22-18 Adding a Mobile Access resource

3. As illustrated in Figure 22-19, right-click on the created Mobile access resource, and select the option **Add -> Mobile network interface** to add the MNI configuration in the subordinate node.

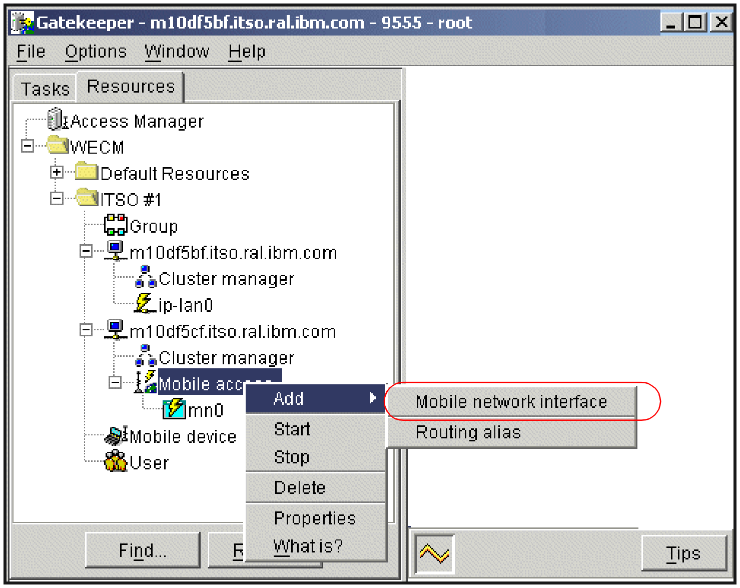
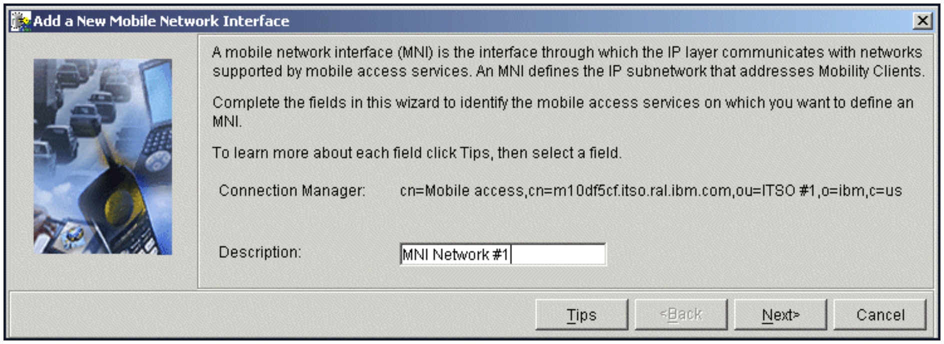


Figure 22-19 Selecting option to add a Mobile Network Interface (MNI)

4. Provide a description and click **Next**.



A mobile network interface (MNI) is the interface through which the IP layer communicates with networks supported by mobile access services. An MNI defines the IP subnetwork that addresses Mobility Clients.

Complete the fields in this wizard to identify the mobile access services on which you want to define an MNI.

To learn more about each field click Tips, then select a field.

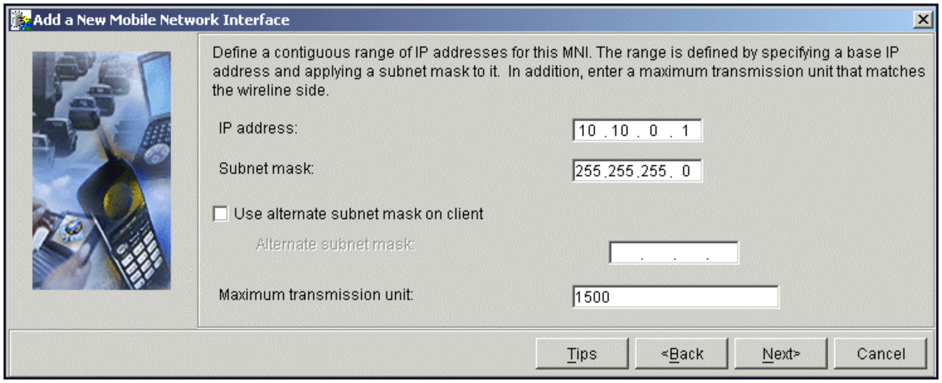
Connection Manager: cn=Mobile access,cn=m10df5cf.itso.ral.ibm.com,ou=ITSO #1,o=ibm,c=us

Description: MNI Network #1

Figure 22-20 Mobile Network Interface description

5. Enter the IP address and subnet mask for the contiguous range of IP addresses in this MNI. Mobility Clients will receive and IP address from this range of addresses. See Figure 22-21.

   Note: If required for your scenario, enter an alternate subnet mask and a maximum transmission unit for the Mobility Clients. Default values are used in the sample scenario described in this chapter.


Add a New Mobile Network Interface

Define a contiguous range of IP addresses for this MNI. The range is defined by specifying a base IP address and applying a subnet mask to it. In addition, enter a maximum transmission unit that matches the wireline side.

IP address: 10 .10 . 0 . 1

Subnet mask: 255 .255 .255 . 0

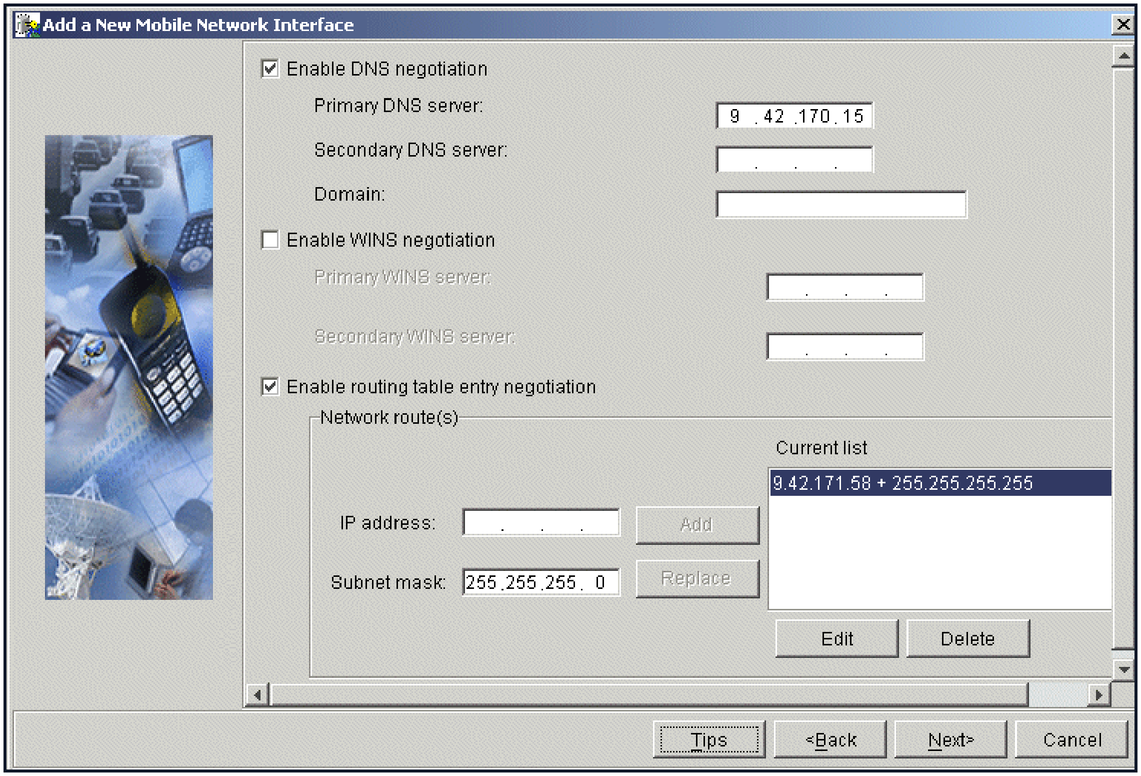
☐ Use alternate subnet mask on client

Alternate subnet mask: . . .

Maximum transmission unit: 1500

Figure 22-21 Mobile Network Interface configuration values

6. Configure a Domain Name Service (DNS), WINS, and routing table negotiation as required for your network.


Add a New Mobile Network Interface

☑ Enable DNS negotiation

Primary DNS server: 9 .42 .170 .15

Secondary DNS server: . . .

Domain:

☐ Enable WINS negotiation

Primary WINS server: . . .

Secondary WINS server: . . .

☑ Enable routing table entry negotiation

Network route(s)

IP address: . . .

Subnet mask: 255 .255 .255 . 0

Current list

9.42.171.58 + 255.255.255.255

Figure 22-22 DNS, WINS and routing table negotiation options

In this scenario the following configuration values are used:

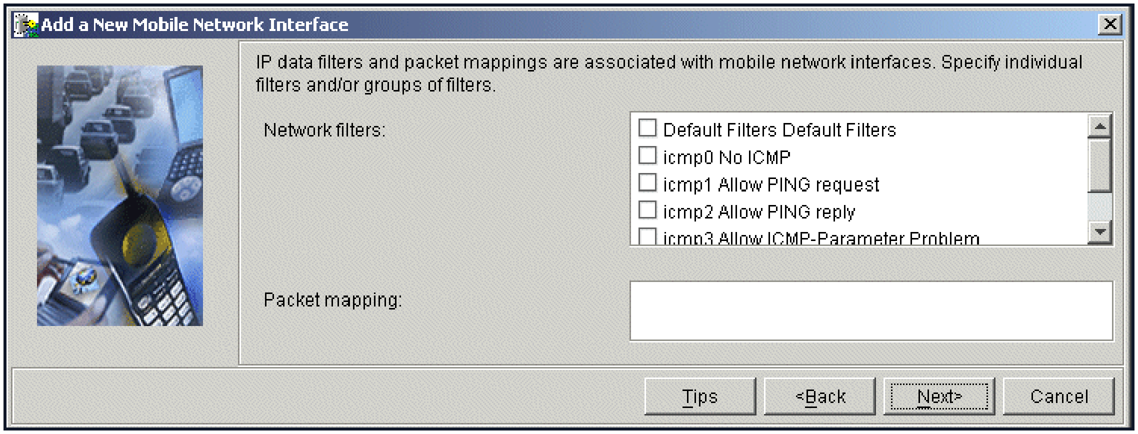
a. Enable the Domain Name Service (DNS) negotiation. Check the **Enable DNS negotiation** option and fill in the primary DNS server, optionally a secondary DNS server and domain as provided by your network administrator.

b. Enable WINS negotiation. (This is not required in this scenario.)

c. Enable routing table entry negotiation. (Required.) A routing table entry is configured for the Mobility Client. In this scenario a route table entry for a single host is entered (subnet mask is 255.255.255.255); however, in a general case you will specify a route table entry for a subnet by using the proper mask. In this scenario the host address 9.42.171.58 is the address of the application server supporting ping (for testing purposes), HTTP, and FTP protocols.

   Note: The routing table entries you provide will be downloaded to the Mobility Clients when connecting to this MNI. The IP addresses and optional subnet masks are the valid routes through which data is forwarded by this MNI.

7. Specify any optional filters and packet mapping you need for this connection (see Figure 22-23). They are not required for this scenario.



Add a New Mobile Network Interface

IP data filters and packet mappings are associated with mobile network interfaces. Specify individual filters and/or groups of filters.

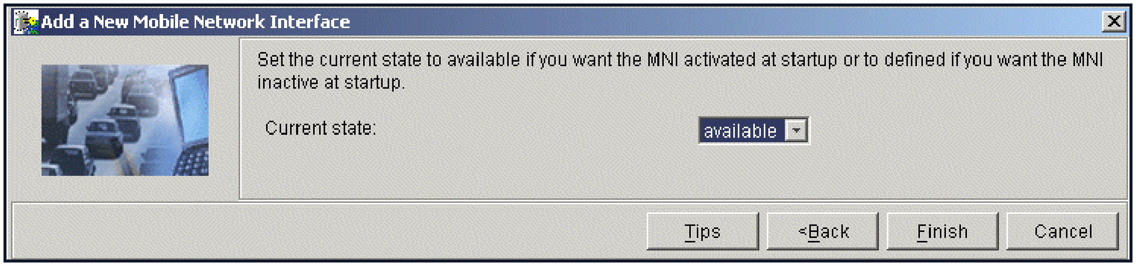
Network filters:

- □ Default Filters Default Filters
- □ icmp0 No ICMP
- □ icmp1 Allow PING request
- □ icmp2 Allow PING reply
- □ icmp3 Allow ICMP-Parameter Problem

Packet mapping:

Figure 22-23 Optional filters and packet mappings

8. Make sure the MNI is available at startup time as shown in Figure 22-24.


Add a New Mobile Network Interface

Set the current state to available if you want the MNI activated at startup or to defined if you want the MNI inactive at startup.

Current state: available

Figure 22-24 Mobile Network Interface (MNI) current state configuration

22.4 Application server requirements

Assuming that your enterprise applications are all running properly, the only requirement is that you will need to add a route to access the subordinate node as specified in its Mobile Network Interface (MNI) configuration. The MNI configuration is illustrated in Figure 22-21 on page 490; it shows that the Mobility Client network is 10.10.0.1 and subnet mask 255.255.255.0

Therefore, for this sample scenario the following network route will be required in the application server machine where 9.42.171.62 is the IP address of the subordinate node:

```
route add 10.10.0.0 mask 255.255.255.0 9.42.171.62 metric 30
```

Note: In this sample scenario the application server and the subordinate node with IP address 9.42.171.62 are in the same subnet. Therefore, this is the only type of route entry required. However, if your application server is not in the same subnet, you will need to add extra routes in the router unless default routes are used. For details about your network you should check with your network administrator.

22.5 Mobility Client configuration

This scenario considers that the Mobility Client has been already installed in a Windows system (Win32), and therefore this section describes the creation of a new connection only:

1. Open the **Mobility Connections** folder.

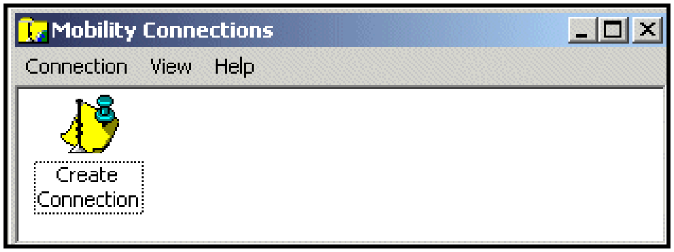
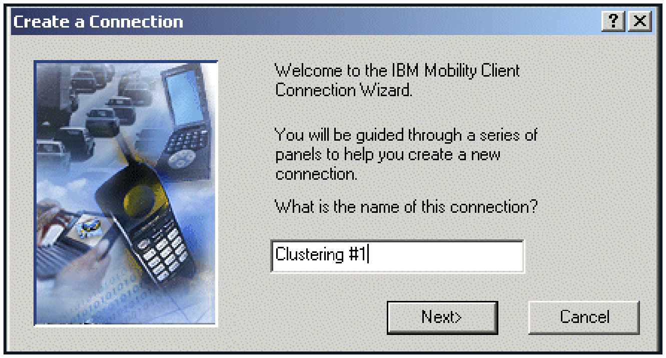


Figure 22-25 Create a Mobility Client connection

2. Double click the **Create Connection** icon, enter the connection name, and click **Next**.



Welcome to the IBM Mobility Client Connection Wizard.

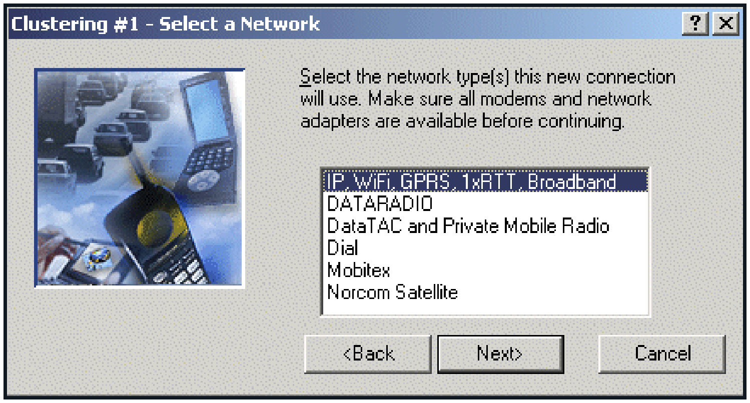
You will be guided through a series of panels to help you create a new connection.

What is the name of this connection?

Clustering #1

Figure 22-26 Connection name

3. Select a network. In this scenario the option **IP, WiFi, GPRS, 1xRTT, Broadband** network option is used as shown in Figure 22-27. Click **Next**.

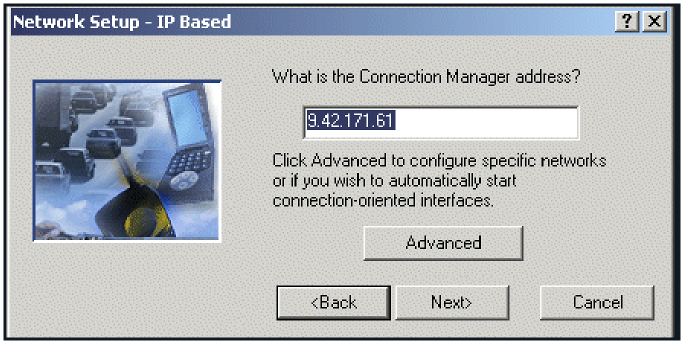


Select the network type(s) this new connection will use. Make sure all modems and network adapters are available before continuing.

IP, WiFi, GPRS, 1xRTT, Broadband
DATARADIO
DataTAC and Private Mobile Radio
Dial
Mobitex
Norcom Satellite

Figure 22-27 Selecting the network type

4. Enter the Connection Manager principal node IP Address and click **Next**.



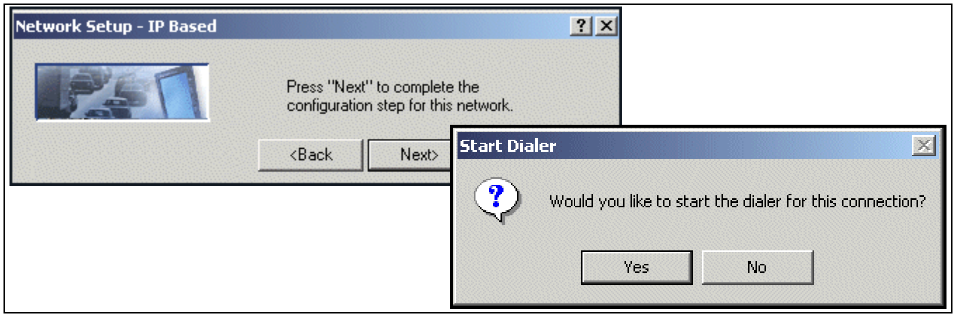
What is the Connection Manager address?

9.42.171.61

Click Advanced to configure specific networks or if you wish to automatically start connection-oriented interfaces.

Figure 22-28 Connection Manager principal node IP address

5. Click **Next** to complete the connection configuration. A connection dialer is not used in this scenario.



Press "Next" to complete the configuration step for this network.

Would you like to start the dialer for this connection?

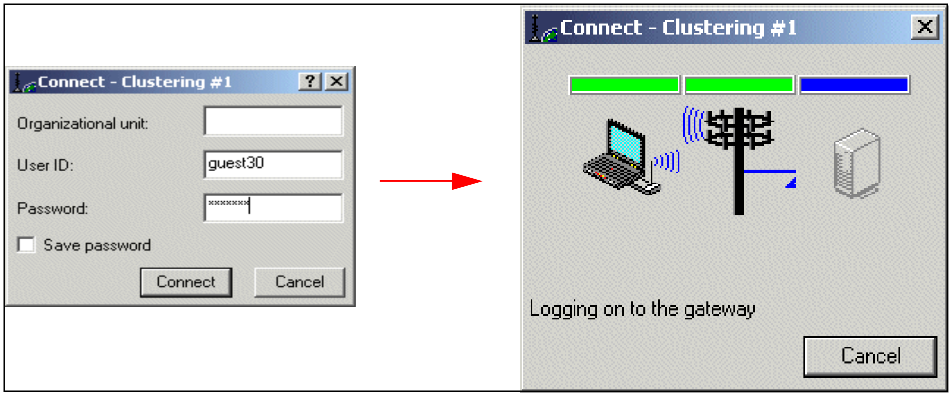
Figure 22-29 Create a new connection

22.6 Running and troubleshooting the scenario

A simple check can be performed to make sure the basic clustering functions are working properly by verifying that an IP address from the subordinate node MNI IP address range has been assigned to the Mobility Client. In a Windows Mobility Client machine this can be done by executing the **`ipconfig`** command.

For example execute the following tasks:

1. In the Mobility Connections folder, double click the connection and enter the user ID and password. Click **Connect** to start the connection to the principal node. A monitor window appears and an icon is created on the toolbar.



Organizational unit:

User ID: guest30

Password: *******

□ Save password

Logging on to the gateway

Figure 22-30 Starting the Mobility Client connection

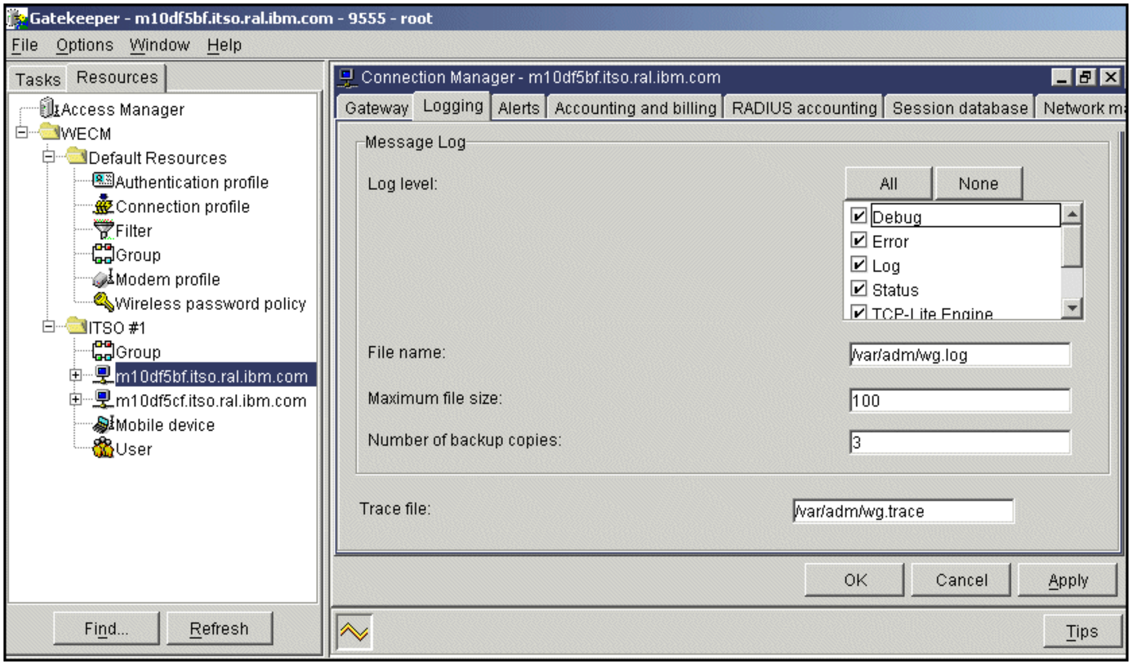
2. When the connection is established, open a command window and type the **`ipconfig`** command. The result should be similar to the one shown in Example 22-1. In this case, we see the following:
   a. Address 10.10.0.4 was obtained from MNI in the subordinate node. It is clear that with the properly route configured in the subordinate node MNI, the Mobility Client applications will use this interface.
   b. Address 192.168.1.105 was obtained from the wireless access point, and it is the transport for the Mobility Client traffic.

Example 22-1 IP configuration

```
Windows 2000 IP Configuration

Ethernet adapter {145CC298-C667-4B76-95D2-9BE6CD321B46}:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 10.10.0.4
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . :

Ethernet adapter Wireless Network Connection:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.1.105
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.1.1
```

3. For testing purposes you may want to activate logging and tracing in Connection Manager. For example, right-click on a Connection Manager

resource, select **Properties**, and click the **Logging** tab. Click **All**, then **Apply**, and finally click **OK**.



Log level:
☑ Debug
☑ Error
☑ Log
☑ Status
☑ TCP-Lite Engine

File name: /var/adm/wg.log

Maximum file size: 100

Number of backup copies: 3

Trace file: /var/adm/wg.trace

Figure 22-31 Enable logging

4. It is always recommended to synchronize the date and time in all Connection Manager machines involved in the clustering scenario. For testing purposes it is also a good practice to stop all the Connection Manager machines, clear the log files, and restart the Connection Manager machines again. This synchronization makes debugging easier when required.
5. If already started, stop the Mobility Client and check the network adapters. The MNI address should not be listed as shown in Example 22-2.

Example 22-2 IP configuration before Mobility Client is started

```
C:\>ipconfig

Windows IP Configuration

Ethernet adapter Wireless Network Connection:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.1.105
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.1.1
```

6. Check the routing table. The NMI routing table should not be listed either since the Mobility Client is not running. For example, see the routes in Example 22-3.

Example 22-3 routing table before Mobility Client is started

```
C:\>route print
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      192.168.1.1   192.168.1.105       20
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1       1
      192.168.1.0    255.255.255.0    192.168.1.105   192.168.1.105       20
    192.168.1.105  255.255.255.255        127.0.0.1       127.0.0.1       20
    192.168.1.255  255.255.255.255    192.168.1.105   192.168.1.105       20
        224.0.0.0        240.0.0.0    192.168.1.105   192.168.1.105       20
  255.255.255.255  255.255.255.255    192.168.1.105   192.168.1.105       1
  255.255.255.255  255.255.255.255    192.168.1.105               4       1
  255.255.255.255  255.255.255.255    192.168.1.105               2       1
Default Gateway:       192.168.1.1
===========================================================================
Persistent Routes:
  None
```

7. Start the Mobility Client and check the new network address, and routing table again. The MNI address (10.10.0.4 in this scenario) should be listed at this time as illustrated in Example 22-4.

Example 22-4 IP configuration after Mobility Client is started

```
C:\>ipconfig

Windows IP Configuration

Ethernet adapter {145CC298-C667-4B76-95D2-9BE6CD321B46}:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 10.10.0.4
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . :

Ethernet adapter Wireless Network Connection:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.1.105
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.1.1
```

8. Check also the new routing table entries. It should see Mobility Client route to access the application server (9.42.171.58) and the route to access the principal node (9.42.171.61). Both route entries are downloaded by Connection Manager.

Example 22-5   routing table after Mobility Client is started

```
C:\>route print
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      192.168.1.1   192.168.1.105       20
      9.42.171.58  255.255.255.255        10.10.0.1         10.10.0.4       1
      9.42.171.61  255.255.255.255      192.168.1.1   192.168.1.105       1
        10.10.0.0    255.255.255.0        10.10.0.4         10.10.0.4       50
        10.10.0.4  255.255.255.255        127.0.0.1         127.0.0.1       50
   10.255.255.255  255.255.255.255        10.10.0.4         10.10.0.4       50
        127.0.0.0        255.0.0.0        127.0.0.1         127.0.0.1       1
      192.168.1.0    255.255.255.0    192.168.1.105   192.168.1.105       20
    192.168.1.105  255.255.255.255        127.0.0.1         127.0.0.1       20
    192.168.1.255  255.255.255.255    192.168.1.105   192.168.1.105       20
        224.0.0.0        240.0.0.0        10.10.0.4         10.10.0.4       50
        224.0.0.0        240.0.0.0    192.168.1.105   192.168.1.105       20
  255.255.255.255  255.255.255.255        10.10.0.4                 4       1
  255.255.255.255  255.255.255.255        10.10.0.4         10.10.0.4       1
  255.255.255.255  255.255.255.255    192.168.1.105   192.168.1.105       1
Default Gateway:       192.168.1.1
===========================================================================
Persistent Routes:
  None
```

9. By adding and deleting the application server route entry, you will enable or disable the access to the FTP or HTTP servers. This can be useful for testing purposes.

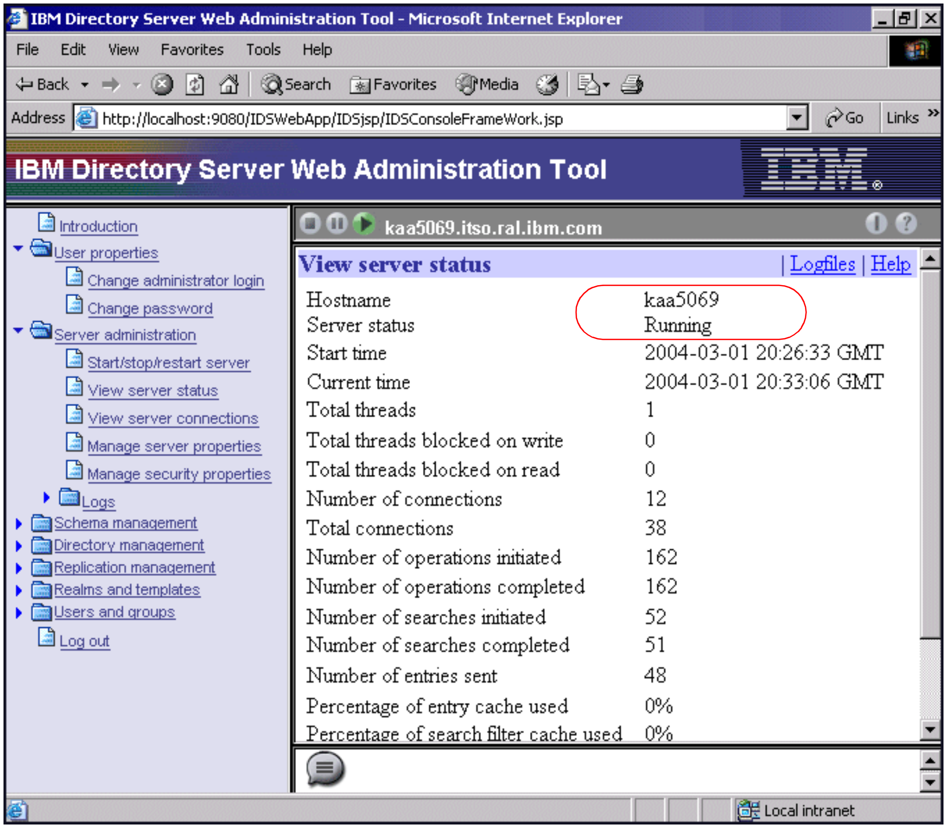
10. The wg.log file in Connection Manager shows the traffic. For example, see the transactions on the subordinate node shown in Example 22-6.

Example 22-6   Sample log entries (wg.log) in subordinate node

```
6290: 4370 (Mar 02 04/16:53:41.8424):WLP::deliver: (entry)
6290: 4370 (Mar 02 04/16:53:41.8425):Deliver: 9.42.171.58:0 -> 10.10.0.4:0 (60)
0000:  45 00 00 3c 1a f6 00 00 7f 01 62 59 09 2a ab 3a
0010:  0a 0a 00 04 00 00 7c 5a 02 00 d7 01
```

22.6.1 LDAP considerations

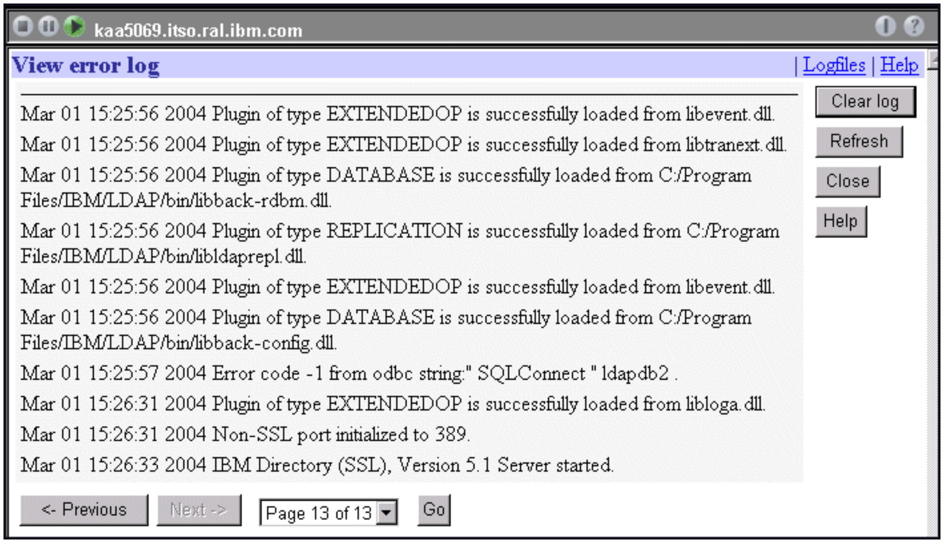
It is essential that LDAP should be up and running. There are several ways to verify this, for example, in this scenario the LDAP Web Administration Tool is used to display LDAP status as illustrated in Figure 22-32.


IBM Directory Server Web Administration Tool - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost:9080/IDSWebApp/IDSjsp/IDSConsoleFrameWork.jsp
Go Links
IBM Directory Server Web Administration Tool
IBM
Introduction
User properties
Change administrator login
Change password
Server administration
Start/stop/restart server
View server status
View server connections
Manage server properties
Manage security properties
Logs
Schema management
Directory management
Replication management
Realms and templates
Users and groups
Log out
kaa5069.itso.ral.ibm.com
View server status
| Logfiles | Help
Local intranet

Hostname	kaa5069
Server status	Running
Start time	2004-03-01 20:26:33 GMT
Current time	2004-03-01 20:33:06 GMT
Total threads	1
Total threads blocked on write	0
Total threads blocked on read	0
Number of connections	12
Total connections	38
Number of operations initiated	162
Number of operations completed	162
Number of searches initiated	52
Number of searches completed	51
Number of entries sent	48
Percentage of entry cache used	0%
Percentage of search filter cache used	0%

Figure 22-32 LDAP status

You can also search for problems in the associated log files as shown in Figure 22-33.



```
Mar 01 15:25:56 2004 Plugin of type EXTENDEDOP is successfully loaded from libevent.dll.
Mar 01 15:25:56 2004 Plugin of type EXTENDEDOP is successfully loaded from libtranext.dll.
Mar 01 15:25:56 2004 Plugin of type DATABASE is successfully loaded from C:/Program Files/IBM/LDAP/bin/libback-rdbm.dll.
Mar 01 15:25:56 2004 Plugin of type REPLICATION is successfully loaded from C:/Program Files/IBM/LDAP/bin/libldaprepl.dll.
Mar 01 15:25:56 2004 Plugin of type EXTENDEDOP is successfully loaded from libevent.dll.
Mar 01 15:25:56 2004 Plugin of type DATABASE is successfully loaded from C:/Program Files/IBM/LDAP/bin/libback-config.dll.
Mar 01 15:25:57 2004 Error code -1 from odbc string:" SQLConnect " ldapdb2 .
Mar 01 15:26:31 2004 Plugin of type EXTENDEDOP is successfully loaded from libloga.dll.
Mar 01 15:26:31 2004 Non-SSL port initialized to 389.
Mar 01 15:26:33 2004 IBM Directory (SSL), Version 5.1 Server started.
```

Figure 22-33 LDAP logging

Note: Other LDAP log files such as the DB2 log and administration log can also help you to troubleshoot problems.

22.6.2 Connection Manager logging

There are several log files in Connection Manager. In this scenario the wg.log file located in the /var/adm directory is probably the most important one to look for problems. For example, Example 22-7 shows a few transactions in the principal node (Cluster Management Protocol transactions).

Example 22-7 Sample wg.log file

```
14198: 1543 (Mar 01 04/14:47:22.1258):TcpPort::read: (return), rc=8
14198: 1543 (Mar 01 04/14:47:22.1259):VERSION_DATA (CMP V2) --
cn=m10df5cf.itso.ral.ibm.com,ou=ITSO #1,o=ibm,c=us
14198: 1543 (Mar 01 04/14:47:25.0625):TcpPort::read: (entry)
14198: 1543 (Mar 01 04/14:47:25.0626):TcpPort: received data (10)
0000:  00 08 04 02 00 04 02 00 00 00

14198: 1543 (Mar 01 04/14:47:25.0627):TcpPort::read: (return), rc=8
14198: 1543 (Mar 01 04/14:47:25.0628):START_DATA_FLOW --
cn=m10df5cf.itso.ral.ibm.com,ou=ITSO #1,o=ibm,c=us
14198: 1543 (Mar 01 04/14:47:27.2583):TcpPort::read: (entry)
14198: 1543 (Mar 01 04/14:47:27.2584):TcpPort: received data (53)
0000:  00 33 04 02 00 02 00 f9 b0 00 00 01 00 00 00 00
0010:  00 00 00 00 00 00 00 00 00 00 30 2e 31 39 30 36
```

```
0020:  37 34 20 30 2e 30 38 31 35 35 38 20 30 2e 30 34
0030:  31 38 34 30 00

14198: 1543 (Mar 01 04/14:47:27.2585):TcpPort::read: (return), rc=51
14198: 1543 (Mar 01 04/14:47:27.2586):LOAD_INFO --
cn=m10df5cf.itso.ral.ibm.com,ou=ITSO #1,o=ibm,c=us
14198:  258 (Mar 01 04/14:47:35.0082):MNCSystem::monitor: (entry)
14198:  258 (Mar 01 04/14:47:35.0083):ip-lan0::monitor: (entry)
14198:  258 (Mar 01 04/14:47:35.0084):ip-lan0::monitor: (return), rc=0
14198:  258 (Mar 01 04/14:47:35.0084):MNCSystem::monitor: (return), rc=0
14198: 1543 (Mar 01 04/14:47:42.0441):TcpPort::read: (entry)
14198: 1543 (Mar 01 04/14:47:42.0441):TcpPort: received data (53)
```

23

Implementing HACMP with Connection Manager

In order to achieve a highly available Connection Manager environment, high availability solutions should be considered for each system component (such as the network, firewall, Connection Manager, etc.). This chapter discusses how to improve the availability of a Connection Manager environment by using IBM's High Availability Cluster Multi-Processing for AIX (HACMP) to provide a failover mechanism for the key Connection Manager components.

This chapter discusses the following:

- Overview of key HACMP concepts
- Alternative configurations using HACMP with Connection Manager
- Installation tips

23.1 Overview of HACMP concepts

This section covers only some key HACMP concepts referenced in this chapter. For more detailed information on HACMP concepts, please refer to *High Availability Cluster Multi-Processing for AIX Concepts and Facilities Guide,* SC23-4864.

23.1.1 Failover

The HACMP software allows a cluster to continue to provide services even though a key system component is no longer available. When a component becomes unavailable, HACMP is able to detect that loss and shift the workload to another component in the cluster. This workload failover capability helps to ensure that none of the nodes in the cluster become a single point of failure. HACMP's failover capabilities can be used to improve system availability during planned maintenance and configuration changes, as well as during unplanned component outages.

HACMP implements a heartbeat mechanism between two AIX servers. When no heartbeat is detected for a pre-defined number of beats, a set of administrator-defined scripts are run and the surviving server takes on the failed server's workload. Since the surviving server takes on the original server's IP address in addition to its own, no client or network changes are required to direct new requests to the surviving server.

23.1.2 Configuration models

HACMP supports two basic configuration models:

- *Mutual takeover* - Where a set of servers share the workload, leveraging all the available hardware and maximizing throughput.
- *Idle standby* - Where one node is designated as the primary node to run the service, and an extra node waits as a hot standby for the other nodes in the cluster.

While the mutual takeover configuration requires less hardware, a significant performance impact may be noticed when failover is invoked and the remaining node is forced to handle both the nodes' workload. Therefore, when selecting the appropriate mode, performance requirements must be considered in addition to the Connection Manager availability requirements.

23.1.3 Application Server

Applications are managed by defining *application server* cluster resources to HACMP. The application server resource includes application start and stop scripts that are provided by the application administrator. HACMP uses these scripts when the application needs to be brought online or offline on a particular node. Example 23-1 illustrates a sample Connection Manager start script.

Example 23-1 Sample Connection Manager start script

```
...
#!/bin/ksh
# /usr/local/bin/start_ewg.ksh

export BANNER="####################################################"
export HA_NODE_NAME=`/usr/sbin/cluster/utilities/get_local_nodename`

echo "\n${BANNER}"
echo "  PRODUCTION APPLICATION START SCRIPT BEGINNING"
echo "${BANNER}\n"

# start HTTP Server
start_http()
{
if [ -x /usr/HTTPServer/bin/apachectl ]
 then
  echo "\nStarting HTTP Server"
  cd /usr/HTTPServer/bin
  ./apachectl start
fi
}

# start DB2 Instance 1
start_wgdb()
{
su - wgdb <<EOF
db2start
EOF
}

# start DB2 Instance 2
start_ldapdb2()
{
su - ldapdb2 <<EOF
db2start
EOF
}

# Start LDAP
```

```
start_ldap()
{
/usr/bin/slapd
}

# Start EWG
start_ewg()
{
/etc/rc.wgated
}

### Start Here
echo "\nStarting HTTP on ${HA_NODE_NAME}"
start_http
echo "\nStarting DB2 on ${HA_NODE_NAME}"
start_wgdb
start_ldapdb2
echo "\nStarting LDAP on ${HA_NODE_NAME}"
start_ldap
echo "\nStarting EWG on ${HA_NODE_NAME}"
start_ewg

echo "\n${BANNER}"
echo "  PRODUCTION APPLICATION START SCRIPT COMPLETED"
echo "${BANNER}\n"

exit 0
```

Example 23-2 illustrates a sample Connection Manager stop script.

Example 23-2 Sample Connection Manager stop script

```
#!/bin/ksh
# /usr/local/bin/stop_ewg.ksh

export BANNER="####################################################"
export HA_NODE_NAME=`/usr/sbin/cluster/utilities/get_local_nodename`

echo "\n${BANNER}"
echo "  PRODUCTION APPLICATION STOP SCRIPT BEGINNING"
echo "${BANNER}\n"

# script to properly shutdown db2 database

# shutdown EWG
stop_ewg()
{
```

```
stopsrc -s wgated
}

# shutdown slapd
stop_ldap()
{
for proc in `ps -ef | grep slapd | grep -v grep | awk '{print $2}'`
do
 echo "Killing slapd daemon"
 kill -9 $proc
done
}

# shutdown DB2 Instance 1
stop_ldapdb2()
{
su - ldapdb2 <<EOF
db2stop force
EOF
}

# shutdown DB2 Instance 2
stop_wgdb()
{
su - wgdb <<EOF
db2stop force
EOF
}

# shutdown HTTP
stop_http()
{
if [ -x /usr/HTTPServer/bin/apachectl ]
 then
  echo "\nStarting HTTP Server"
  cd /usr/HTTPServer/bin
  ./apachectl stop
fi
}

# Start Here
echo "Shutting Down EWG on ${HA_NODE_NAME}"
stop_ewg
echo "Shutting Down LDAP on ${HA_NODE_NAME}"
stop_ldap
echo "Shutting Down DB2 on ${HA_NODE_NAME}"
stop_wgdb
stop_ldapdb2
echo "Shutting Down HTTP on ${HA_NODE_NAME}"
```

```
stop_http


echo "\n${BANNER}"
echo "  PRODUCTION APPLICATION STOP SCRIPT COMPLETED"
echo "${BANNER}\n"

exit 0
```

23.1.4 Service IP addresses

A service IP address is an IP address over which services such as an application are provided, and over which the client communicates. In the event of a node failure, HACMP can transfer the service IP address to another node in the cluster. By using the same IP service address for multiple nodes, the user can continue to access the service after a node failover without any user reconfiguration.

23.2 HACMP sample configurations

In Chapter 22, “Clustering” on page 473 the use of clustered subordinate nodes is described as a means for improving the availability of the Connection Manager environment. However, in order to achieve true high availability, the Connection Manager principal node, relational database, and LDAP servers (for a basic configuration) should also be clustered.

23.2.1 High availability of the principal nodes

The principal node in a Connection Manager cluster is a single point of failure. Using HACMP, principal nodes can be clustered to eliminate that single failure point, as indicated in Figure 23-1

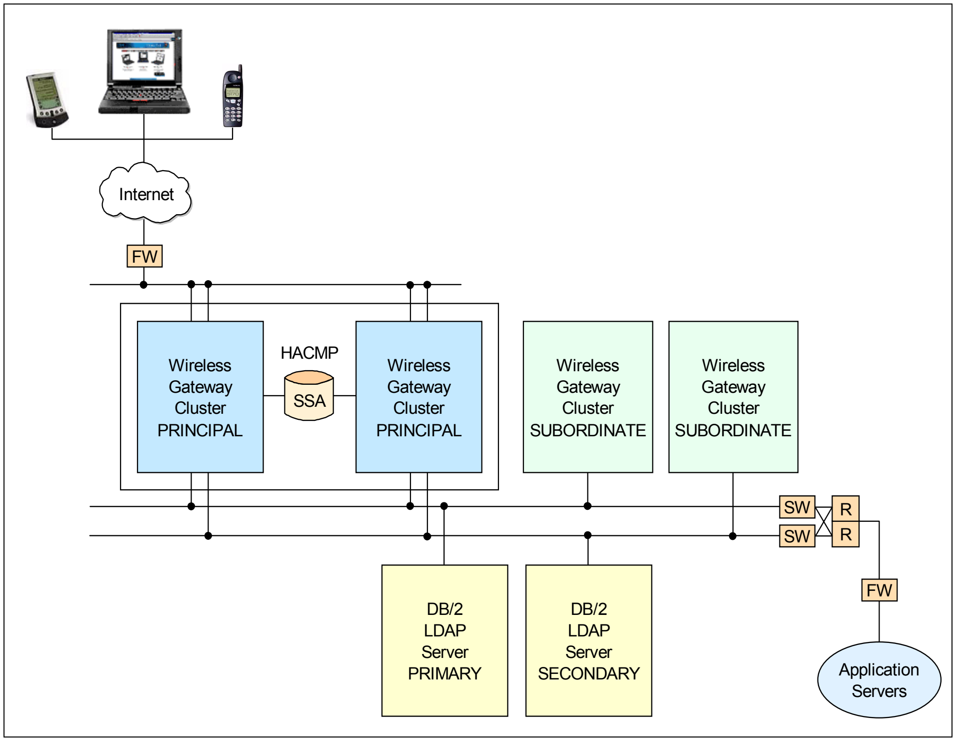

Internet
FW
Wireless Gateway Cluster PRINCIPAL
HACMP
SSA
Wireless Gateway Cluster PRINCIPAL
Wireless Gateway Cluster SUBORDINATE
Wireless Gateway Cluster SUBORDINATE
SW
SW
R
R
FW
DB/2 LDAP Server PRIMARY
DB/2 LDAP Server SECONDARY
Application Servers

Figure 23-1 Using HACMP to cluster Connection Manager principal nodes

Some notes about Figure 23-1 are:

- In this example we have selected to run an idle standby cluster configuration. The shared disk in the HACMP cluster is for HACMP information and does not contain any Connection Manager user or configuration data.
- HACMP is used to provide backup for the Mobile Network Connections (MNCs) running on the principal nodes. In the event of failure, either primary node will act as the Connection Manager for the entire Connection Manager environment.
- HACMP is not required to provide backup for the subordinate nodes. In the event of failure, either subordinate cluster node supports the load of the failed node.
- There are several options for achieving redundant/backup RDBMS and LDAP servers. These options are discussed in the “High availability of LDAP and DB2” on page 510.

23.2.2 High availability of the subordinate nodes

As stated previously, it is not necessary to define an HACMP cluster for the subordinate nodes since Connection Manager's cluster manager capabilities will handle subordinate node failover if necessary.

Note: Connection Manager Cluster Manager should not be used for messaging services or HTTP Access Services. For these services, an alternate clustering solution should be used.

23.2.3 High availability of LDAP and DB2

With HACMP

Figure Figure 23-2 adds an additional HACMP cluster to our original configuration. The external disk device (SSA in this case) should be mirrored or RAID-configured for data redundancy. By sharing the external disk where the user and configuration data are stored, either Connection Manager principal node can access the data should one node become unavailable. Configuring the HACMP cluster in idle standby mode simplifies the requirements to manage simultaneous update access to the databases.

Note: With this configuration, users will not be able to log on to Connection Manager while the failover is transitioning.

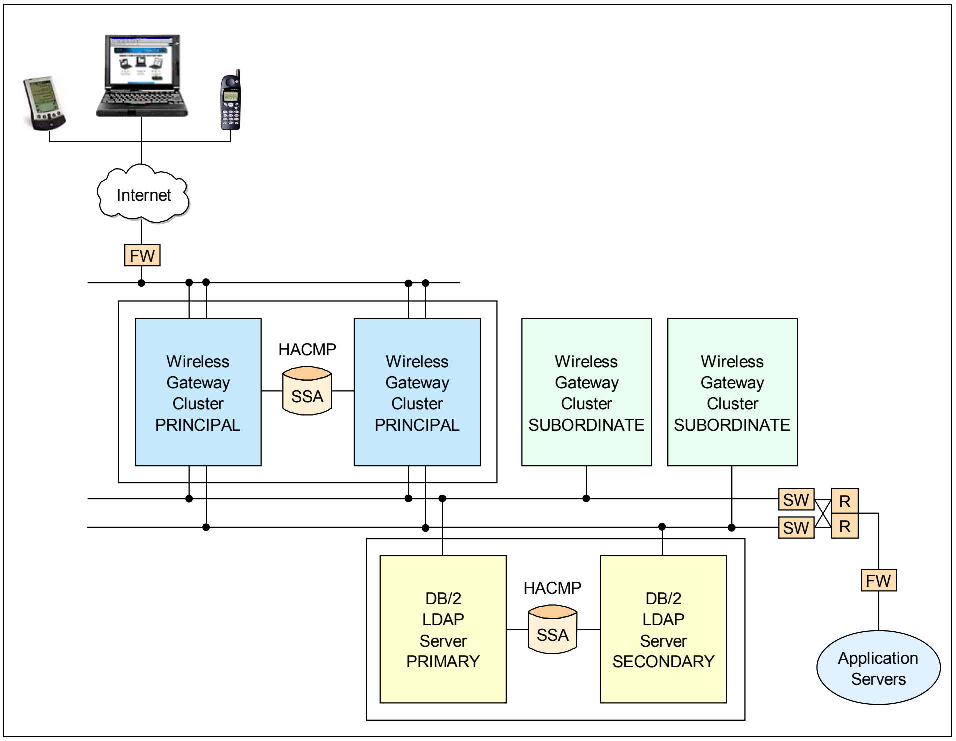

Internet
FW
Wireless Gateway Cluster PRINCIPAL
HACMP
SSA
Wireless Gateway Cluster PRINCIPAL
Wireless Gateway Cluster SUBORDINATE
Wireless Gateway Cluster SUBORDINATE
SW
SW
R
R
FW
DB/2 LDAP Server PRIMARY
HACMP
SSA
DB/2 LDAP Server SECONDARY
Application Servers

Figure 23-2 Using HACMP for DB/2 and LDAP high availability

An alternative approach still using HACMP is to place the DB/2 and LDAP on the same nodes as the Connection Manager Principal node as shown in Figure 23-3. This configuration reduces hardware requirements. Although the principal node is only running the MNC, careful performance planning should be performed before selecting this approach. In addition, the accessibility of directory data (i.e., external access) should be evaluated from a security perspective before selecting this configuration.

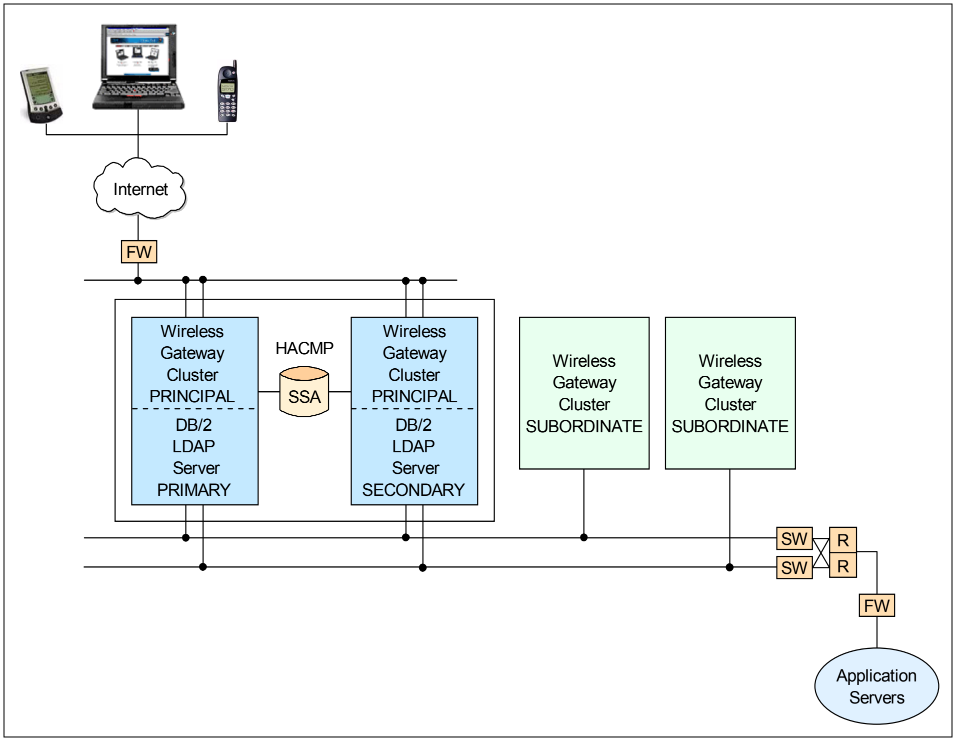

Internet
FW
Wireless Gateway Cluster PRINCIPAL
DB/2 LDAP Server PRIMARY
HACMP
SSA
Wireless Gateway Cluster PRINCIPAL
DB/2 LDAP Server SECONDARY
Wireless Gateway Cluster SUBORDINATE
Wireless Gateway Cluster SUBORDINATE
SW
SW
R
R
FW
Application Servers

Figure 23-3 Using a Single HACMP Cluster for Connection Manager

Without HACMP

If LDAP and DB2 are not placed in an HACMP environment, a method for keeping the primary and secondary servers synchronized must be determined. For more information on using native DB2 Backup and LDAP Replication utilities, please refer to the DB2 and LDAP product installation guides. When using these native product utilities, the process for switching to the secondary server when the primary becomes unavailable is a manual one.

Alternatively for LDAP, you can elect to have Connection Manager manage the “failover” function for LDAP, and optionally the LDAP synchronization through the use of *Secondary Directory Service Server (DSS)*. With this approach, if the primary LDAP server is unavailable, Connection Manager will automatically switch to the secondary LDAP specified. This feature will not provide high availability for DB2.

To specify a secondary DSS, select the appropriate Connection Manager within Gatekeeper. Then right-click, select **Properties,** and log in. Then select the **DSS** tab. If this is an existing Connection Manager, the primary DSS information will already be filled in. Enter the IP address and port of the secondary DSS. Select the type of DSS desired:

- Read Only Replica

  The secondary DSS is configured as a replica. The primary (master) DSS is configured to propagate updates to the secondary DSS. If the primary DSS becomes unavailable, the Access Manager retrieves entries from the secondary DSS, but does not update it. The DSS administrator is responsible for determining the replication schedule between the primary and secondary servers. Select this option if you do not want Connection Manager to update the secondary DSS.

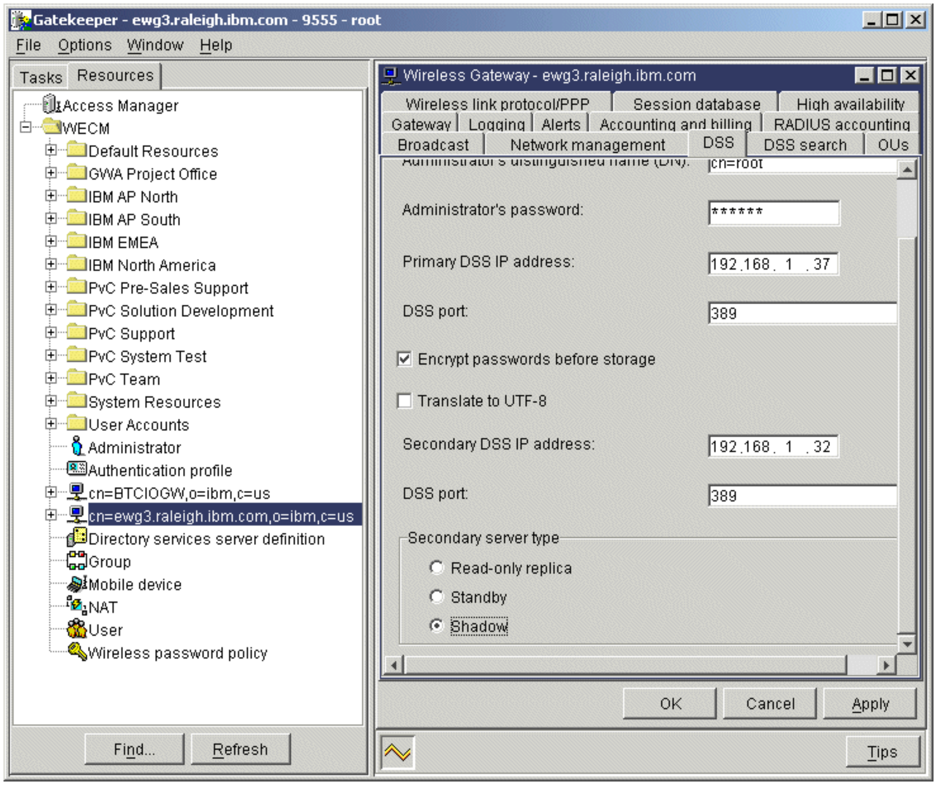
  > **Important:** Since Connection Manager does not write to the secondary DSS with this option, no changes to users' attributes (password, lockout status, e-mail address, etc) should be made while the secondary DSS is in use. Any changes made during this period will be lost.

- Standby

  Configured as a stand-alone DSS, the secondary DSS is used only when the primary DSS is unavailable. The DSS administrator is responsible for maintaining the database synchronization between the primary and secondary directory service servers. If the primary DSS becomes unavailable, the Access Manager retrieves entries from the secondary DSS, and will make updates to it. Select this option if in general you want to control synchronization of the replicas, but also want users and administrators to be able to make changes while the primary DSS is unavailable. Additionally, this option provides a known point from which to recover when the failed database comes back on-line.

- Shadow

  Configured as a stand-alone DSS, the primary and secondary DSSs are simultaneously updated by the Access Manager. If the primary or secondary DSS becomes unavailable, the Access Manager does not update that server. When that DSS becomes available again, the DSS administrator is responsible for synchronizing the database between the primary and secondary servers. Select this option if you want Connection Manager to manage the database synchronization when both servers are available. This option also provides a known point from which to recover when the failed database comes back on-line.



Administrator's distinguished name (DN): cn=root

Administrator's password: ******

Primary DSS IP address: 192.168.1.37

DSS port: 389

☑ Encrypt passwords before storage

□ Translate to UTF-8

Secondary DSS IP address: 192.168.1.32

DSS port: 389

Secondary server type

○ Read-only replica

○ Standby

● Shadow

Figure 23-4 Specifying the secondary DSS

23.3 Installation tips

This section contains installation guidance for configuring HACMP for Connection Manager. It is not intended to be a comprehensive guide to installing HACMP. For more detailed information on HACMP installation, please refer to *High Availability Cluster Multi-Processing for AIX Installation Guide,* SC23-4861.

Sharing the IP service address

After installing HACMP on both cluster nodes and Connection Manager on the first principal node, force a failover to the second principal node before installing Connection Manager on the second principal node. This will allow the two principal nodes to “share” the same service IP address so users will not require configuration changes to access the second server in a failover situation.

Sharing the configuration data

After installing Connection Manager on the second principal node copy the file `/usr/lpp/wgated.conf` from the first node to the second before modifying the second node's configuration with Gatekeeper. This will copy over the basic configuration data so the two nodes are configured the same.

Sharing the modified LDAP schema

After configuring Connection Manager on the first principal node copy the file `/usr/ldap/etc/v3.modifiedschema` from the primary(master) LDAP server to the secondary (replica) LDAP server. This will copy the Connection Manager-modified schema to the replica LDAP so the two nodes are configured the same.

Attention: This file must be copied before any data is imported or replicated to the replica LDAP.

SSL certificates

Ensure that any SSL certificates installed on the first principal node are also installed on the second principal node.

24

Problem identification and resolution

This chapter provides information to help determine and resolve problems you might encourter between the client, and other components interfacing to Connection Manager. When you are re troubleshooting Connection Manager and its interfaces to other components and devices, a systematic approach works best.

An unsystematic approach to troubleshooting can result in wasting valuable time and resources, and can sometimes make symtoms even worse. Define the specific symptoms, identify all potential problems that could be causing the symptoms, and then systematically eliminate each potential problem (from most likely to least likely) until the symptoms disappear.

This chapter only provides problem identification and resolution between Connection Manager and its interfaced components, and a few scenarios of service components internal to Connection Manager. For a detailed understanding of troubleshooting Connection Manager, see the *Connection Manager version 5.0 Administration Guide*, “Chapter 9” in the product InfoCenter documentation.

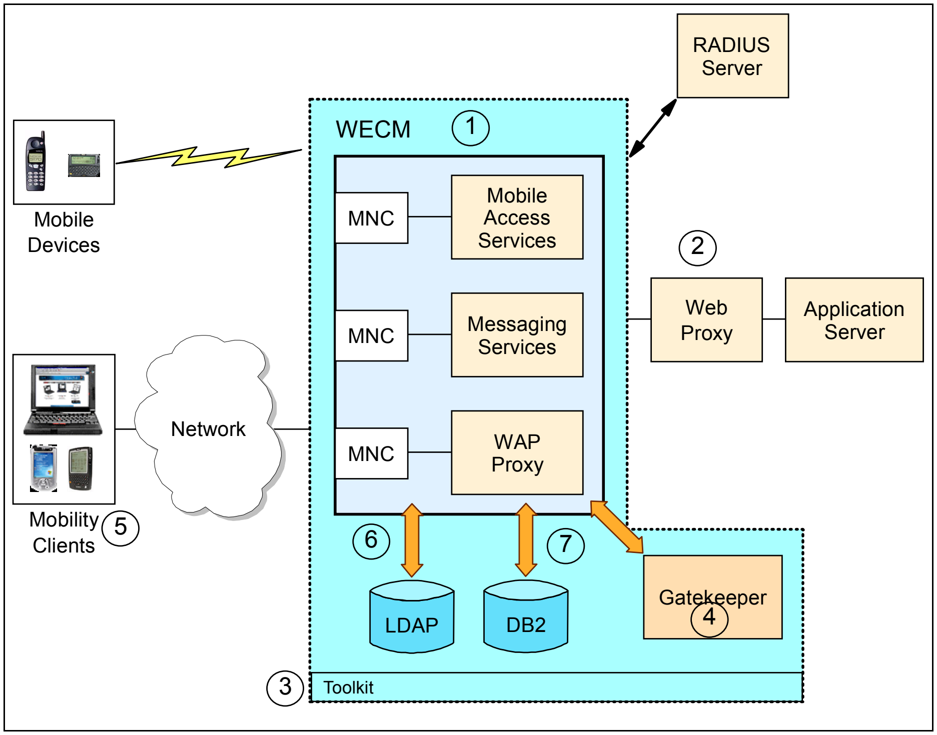


Figure 24-1 Connection Manager components

24.1 Connection Manager installation verification

This is a suggested list of steps to verify your Connection Manager installation. For example:

1. Create users.
2. Verify VPN over LAN is operating (IP-LAN MNC):
   a. Ping the gateway's Ethernet address from the client machine without starting the Connection Manager client.
   b. Ping the MNI address, and ping an enterprise server if available.
   c. Log on to Connection Manager.
   d. Attempt to execute transaction with the customer's application.
   e. Log off of Connection Manager.
   f. Repeat the above steps over the networks you will be using.

3. Verify WAP Client connectivity:
   a. First, use a phone simulator over your LAN.
   b. Execute the transaction using the customer's application with the phone simulator.
   c. Attempt to connect with the actual phone or device using the customer's authentication and connectivity profiles set.
   d. Execute transactions using the customer's application from the actual phone or device.
4. Messaging:
   – Verify the configuration of SMTP MNC for use by the messaging gateway; use mail relay with access to the mail domain where you can actually verify that new mail had been received:
     - Use the sample SMTP program in Chapter 18., "Messaging services" on page 371 to execute SMTP push to mail address.
     - Use a sample program to execute the WAP push to the phone simulator.
     - Use the sample program to send data over the network to the customer device.
   – Verify that the messaging gateway configurable TCP port is open, push initiators listens by default on port 13131:
     - Verify that this port is in listen state; you do this by executing the following command on the Connection Manager server: **`netstat -an |grep`** <configured port**>**
     - When configured to use SMTP MNC, the messaging gateway will send SMTP a "hello" message to the configured relay host. Review the wg.log or SMTP replay host.log to verify this completes successfully.

Common problems if TCP-Lite was installed for optimization

The TCP-Lite Connection Manager service provides a reliable transport for delivering TCP over the air. This service is optional and may not be installed. For example:

- TCP-Lite Connection Manager service gets invoked by a match of source/destination port, and source/destination address in a flow through the VPN.
- The simplest configuration, which is also the default is involved for all ports and addresses.
- If TCP-Lite Connection Manager service was added, but you are not seeing evidence of it existing, check the wg.log.

24.2 HTTP services common problems

Connection Manager HTTP service can function in two modes:

- Standard reverse proxy
- Authentication server

In the reverse proxy mode, HTTP services at least four “hops” in order to complete a transaction. These hops are:

- Requestor
- Connection Manager Service
- HTTP proxy
- HTTP server

Symptom:

All your attempts of sending a request to Connection Manager fail

Action:

- Is the HTTP service defined? Verify by using the Gatekeeper interface to look for the HTTP service object.
- The HTTP server opens two TCP ports (you can verify with the **`netstat -an |grep`** *<port defined for HTTP service>)*. One port should be for HTTPS transactions with the default 443; this should be labeled in the Gatekeeper as “TCP to listen on.” The other, which sends HTTP redirect to the defined HTTPS service port, is labeled `Redirect HTTP ports in Gatekeeper`.

Symptom:

User authentication successful, but no response is returned.

Action:

- Point your browser at the HTTP proxy host and port configured for the HTTP service.
- Verify that the endpoint server, behind the reverse proxy, will return a response.
- Check for TCP connection to the HTTP Proxy by doing a **`netstat -an |grep`** *<proxy’s listen port>*

Common VPN symtoms and resolutions

Symptom:

IP routing seems to be causing problems.

Action:

- The 0th address in a subnet definition is reserved for the network address. Do not place the MNI at this address in the subnet.
- Do not place the MNI at the uppermost address in the subnet definition. This is the broadcast address and is reserved.
- If you use “non routable” addresses for your MNI network such as 192.168.1.x or 10.10.10.x, you must either implement NAT with a routable address for the MNI, or provide the proper route table entries at nodes between the MNI and the target endpoint host.
- You must be able to ping the MNI address from the host endpoint. The MNI definition must fit into the enterprise’s overall IP addressing scheme. The Connection Manager is an IP routing resource and must be treated as such.
- To verify MNI is in an “UP” state, and to review details on its configuration, issue an **`ifconfig -a`** on the gateway machine.
- If unable to connect a client type, make sure the connection port is in the proper state, use **`netstat -a | grep`** *<port #>* to determine state. The following should provide you with some guidance:
  - UDP 8889 is default for IP-LAN VPN MNC.
  - TCP 8888 wireless client change password application
  - TCP 9555, 9559 SSL for gateway
  - UDP 9200, 9201, 9202, 9203 for WAP IPV4 bearer
  - TCP 13131, 13132 SSL, Push Initiator/messaging gateway
  - Use x25status and x25monitor commands if X.25 based MNC’s are used
  - You can reference the completed list in the *Connection Manager 5 Administration Guide*
- If you are unable to connect to PPP or with the wireless clients, and the WAP transaction fails, you will need to make sure the MNI address is the WAP gateway address configured in the WAP application.
- If you suspect the gateway, Gatekeeper, or Access Manager is not running, execute a **`ps -ef | grep`** *<process name>.* The Access Manager only runs when Gatekeeper is connected.

Example 24-1 Connection Manager processes

wgated	Gateway process
wgmgrd	Access Manager process
wgcfg	Gatekeeper process
wgattachd	Gateway watchdog process

On the client side, use the Window’s task list for client processes.

Example 24-2 Mobility Client processes

artcore	Wireless client
artdhcp(NT/Win2K only)	Client dhcp service runs without artcore
artbcast	Default broadcast listener

Toolkit users

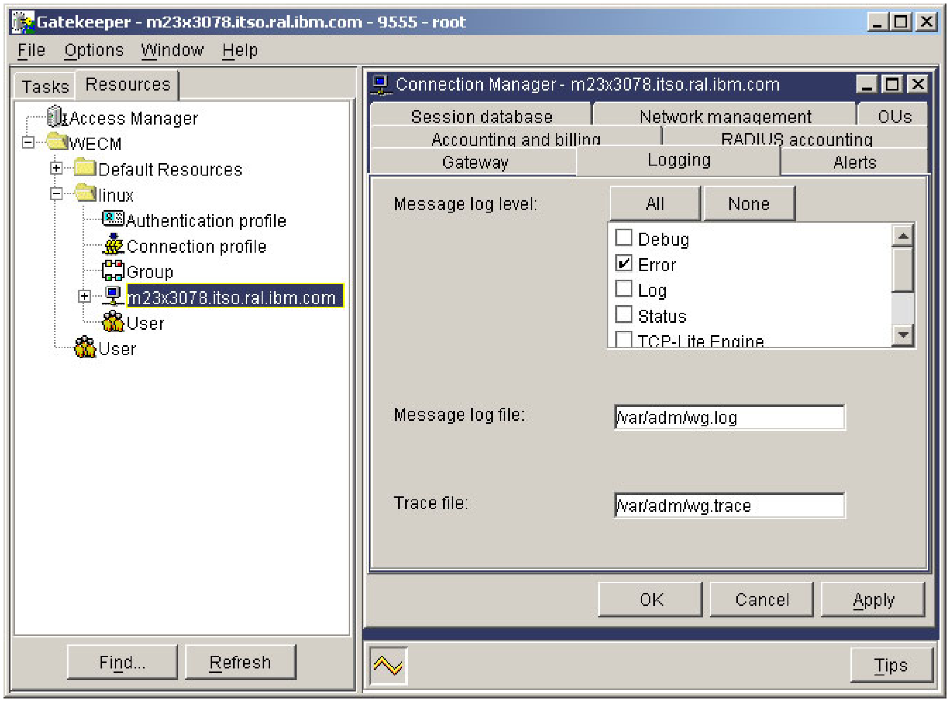
Messaging API users and developers can enable logging for the API by editing the push.properties file. The location of this file will vary depending upon development environment:

- Set **push.tracing=true**/false
- Set **push.tracing.console=true**/false for output to std_out

Wireless client API developers can enable trace for the wireless Mobility Client. Wireless Mobility Client API activity is traced by the `arttrace.exe` program.

24.3 Connection Manager log and trace files

Logging is global across all gateway activities, and trace is specific to a given user. A normal production system should run error and warning levels options.



Gatekeeper - m23x3078.itso.ral.ibm.com - 9555 - root

Connection Manager - m23x3078.itso.ral.ibm.com

Message log level: All / None

☐ Debug
☑ Error
☐ Log
☐ Status
☐ TCP-Lite Engine

Message log file: /var/adm/wg.log

Trace file: /var/adm/wg.trace

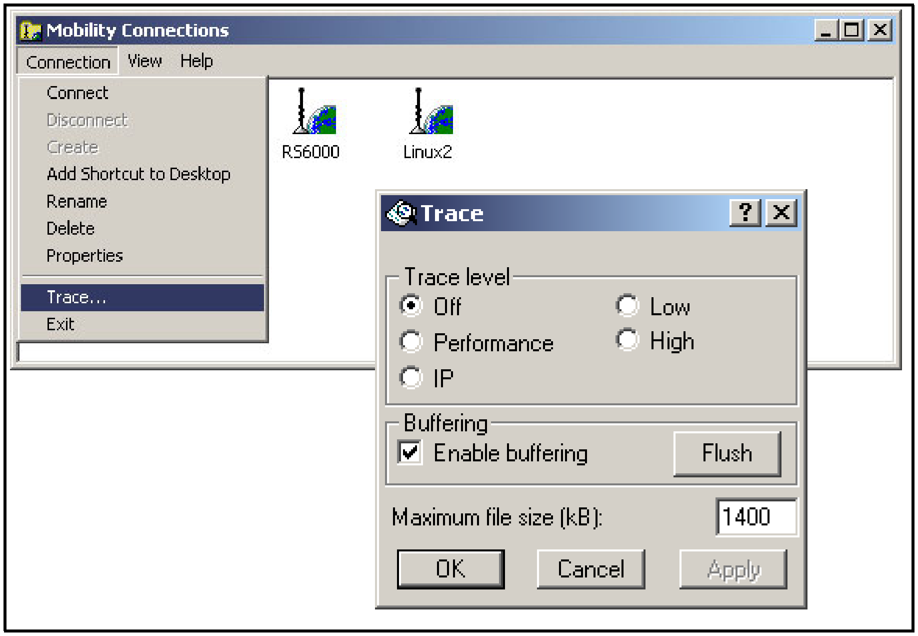
Figure 24-2 Logging and traces

- The gateway log and trace file location and name are configurable using Gatekeeper.
- Gatekeeper:
  - Default location is /var/adm
  - Default file names are wg.log and wg.trace
- Resetting logs and traces generates wg.log.yy.mm.dd.hh.mm.ss files; you can reset by issuing a `chwg -r all` command or by right-clicking the **Gatekeeper Gateway** object.

> **Note:** Always be sure to reset logs. It is recommended that the directory in which log and trace files are is stored is in its own file system. The logging subsystem of Connection Manager checks for available space and automatically reduces log level as the file system nears capacity. An SNMP trap is invoked as log levels are reduced.

24.4 Wireless client trace

The wireless client trace connection GUI is a tool that provides you with a trace log of the Connection Manager activity. If you are having trouble connecting a wireless client, or the applications are not functioning properly over a Connection Manager link, you may enable the trace subsystem.



Trace level
● Off ○ Low
○ Performance ○ High
○ IP

Buffering
☑ Enable buffering

Maximum file size (kB): 1400

Figure 24-3 Mobility connections

It is IP and the WLP layer sampling that allows you to capture limited information from an active gateway:

- The trace file name is <install dirctory>\arttrace.txt.
- The default file limit of the trace file is 1400 bytes on Win32; on Palm it is a percentage of the available storage.
- Once a file reaches the limit a new one is created, and contents of the old file are moved into arttrace2.txt. Only two files of the maximum limit will be maintained.
- Trace levels of “Low” and “Performance” are not used in practice; “High” or “Off” are most common
- If buffering, which is a distinct performance improvement, the recommendation is to leave the arttrace.exe file running, and click the **Flush** button after you are done recreating the problem.

- Leaving trace on is a distinct performance hit to your machine.
- You must use wg_trc program to view trace files, and can be enabled for a given UID from Gatekeeper or the user's property panel.

24.5 Connection Manager and LDAP

Connection Manager communicates with LDAP when:

- wgated process initalizes
- A user's session changes state.
- Gateway configuration changes are made.
- User records are changed and users are added.
- wgated terminates.

Techniques to troubleshoot problems with IBM LDAP:

- Ping the LDAP server from Connection Manager.
- Point HTTP browser <LDAP Host>/ldap and try to log in with the UID/PWD used in the Connection Manager configuration.
- Review <LDAP server>/etc/slapd32.conf to confirm server port, ibm-slapdPort and ibm-slapdErrorLog
- Review ibm-slapdErrorLog, /tmp/slapd.errors by default.
- Connect using an LDAP browser like DMT or Softerra.

24.6 Connection Manager and RDBMS

RDBMS, Connection Manager communicates with a database when:

- wgated process initializes.
- A VPN user session changes state while accounting support is enabled.
- The messaging gateway persists messages for delayed delivery.
- wgated terminates.

Here are some techniques for troubleshooting problems occurring with connectivity with DB2:

1. Ping the DB2 server from Connection Manager.
2. Review <DB2 server>/etc/services to confirm server instance ports, db2c<instanceID> and db2i<instanceID>.
3. On server machine, su -<server instance ID> and invoke DB2 shell.

4. Stop and start DB2.
5. List the database directory and list the node directory.

Part 1

Appendixes

A

AIX: Step-by-step installation

The objective of this chapter is provide a step-by-step script that can be used for Proof of Concept (PoC) reasons. The topics covered in this chapter include:

- Planning
- AIX 5.2 Installation and basic configuration
- DB2 Enterprise Edition V7.2 AIX 32-bit installation
- DB2 v7.2 Fix pack 10 for AIX5 installation
- IBM Directory V4.1 AIX installation and configuration
- IBM HTTP Server V1.3.19.3 for AIX installation
- IBM WebSphere Everyplace Connection Manager Installation
- Gatekeeper installation and configuration

In order to attend exclusively to PoC's demands, this script's information applies when the intention is to install the product and its requisites in the same system. It is supposed that the hardware is dedicated to this Proof of Concept. The WAP resources will not be configured too. The resources described are the VPN and IP networks connection resources.

The software versions utilized to develop this script are not necessarily the minimum versions required for the WebSphere Everyplace Connection Manager v5 installation.

24.7 Planning the Proof of Concept

There are some requirements that must be attended. If all of these requirements are attended, the Proof Of Concept will be successful. This script was obtained from the host used for installation and testing during the production of this chapter. The software versions used to compose it are described in the Software Requirements Session.

Hardware requirements

The hardware must attend at least the minimum of 375 MHz processor and 256 MB RAM available.

The hardware must contain a hard disk with at least 8 Gb of free space.

The hardware must have a network adapter.

System requirements

AIX V5.2

The hostname must be defined and associated with an IP address, and recognized by the DNS server.

Software requirements

To follow this script the software codes must be available. The software versions used to compose this script were:

- DB2 Universal Database Enterprise edition v7.2 AIX 32-bit
- DB2 Fix Pack 10 for AIX 5 (32-bit and 64-bit)
- IBM Directory V4.1 AIX
- IBM HTTP Server V1.3.19.3
- WebSphere Everyplace Connection Manager v5

24.8 Operating system installation

These are the steps:

1. Start the machine.
2. Insert the operating system CD into the drive.
3. Press F1 on startup to enter the system management services.
4. Select the option **Multiboot** and the press Enter.
5. Select the option **Install From**. Press Enter.
6. Select the **CD-ROM device** with the space bar.
7. Go to **Install** option. Press Enter.

8. Select the option **AIX 5.2.0**. Press Enter.
9. The screen will turn to the text mode. Press F1 and then press Enter.
10. Select **1** to select the English language option and press Enter.
11. In the Installation and Maintenance menu, select option **2**. Press Enter.
12. In the Installation and Settings screen, select **1**. Press Enter.
13. In the Change Method of Installation screen, select **1**. Press Enter.
14. In Change Disk where you want to install, accept the default and press Enter.
15. In the Installation and Settings screen, press Enter.
16. In Overwrite Installation Summary, press Enter.
17. The Installation will begin. This process can take several minutes.

24.9 Configuring the operating system

At the end of the installation process, the Configuration Assistant screen will appear:

1. Click **Accept**.
2. Click **Next**.
3. Select **Set or Verify System Date and Time.**
4. Click **Next**.
5. Set the date, time, and region.
6. Select **Set the password for Root user**.
7. Click **Next**.
8. Set the same password in the two fields.
9. Select **Next**.
10. Select **Configure Network Communications (TCP/IP)**.
11. Select **Next**.
12. Select **Manually Configure TCP/IP.**
13. Select **Next**.
14. Set the hostname.
15. Set the IP address.
16. Set subnet mask.
17. Select **Next**.
18. Chose the network interface.
19. Click **Next**.
20. In the option Media Speed, select **Auto-negotiation**.
21. Click **Next**.
22. Set the default gateway address.
23. Set the domain name.
24. Set the IP address of domain name server.
25. Click **Next.**
26. Select **Review the Tasks**.
27. Click **Next**.
28. Click **OK**.

29. Click **Close**.
30. Select **Exit Configuration Assistant**.
31. Click **Next**.
32. Select **Finish now and do not start config assistant when restarting the operating system.**
33. Click **Finish**.
34. Log on as root.

24.10 Pre-installation tasks

The following details the tasks that should be completed to prepare the operating system prior to installing Connection Manager and related products.

24.10.1 Creating CD-ROM file system

Create a CD-ROM filesystem from which to load CDs during installation:

1. `#smitty`
2. Select the option **System Storage Management (Physical & Logical Storage)**.
3. Select the option **File Systems**.
4. Select **Add/Change/Show/Delete File Systems**.
5. Select **CD-ROM File Systems.**
6. Select **Add a CDROM File System.**
7. In the Device Name field, press F4.
8. Select the correct device name (cd0).
9. In Mount Point field insert /cdrom.
10. Press Enter.
11. At the end of the process, press F10.

24.10.2 Adding blocks to main file systems

Be sure that the file system’s free space is at least equal to the one shown in Example 24-3. This result was obtained from the host used for installation and testing during the production of this chapter. To increase the file system size, use the following commands:

1. **`#chfs -a size=+300000 /`**
2. **`#chfs -a size=+4000000 /usr`**
3. **`#chfs -a size=+100000 /var`**
4. **`#chfs -a size=+100000 /tmp`**
5. **`#chfs -a size=+200000 /home`**
6. **`#chfs -a size=+100000 /opt`**

When complete, run the **`df -k`** command to ensure that the file system space is correctly allocated. The result of this last command looks similar to Figure 24-3.

Example 24-3 Results of df -k command

```
# df -k
Filesystem    1024-blocks      Free %Used    Iused %Iused Mounted on
/dev/hd4           196608    175684   11%     1386     2% /
/dev/hd2          2818048   2017352   29%    23563     4% /usr
/dev/hd9var         98304     89444   10%      428     2% /var
/dev/hd3            98304     95168    4%       26     1% /tmp
/dev/hd1           163840    158576    4%       18     1% /home
/proc                   -         -    -         -     -  /proc
/dev/hd10opt        98304     88212   11%      332     2% /opt
#
```

24.11 Installing DB2 Universal Database V7.2

For a sample installation of DB2 Universal Database V7.2, execute the following steps:

1. Insert the DB2 install code in CD ROM drive.
2. `#mount /cdrom`.
3. `#cd /cdrom`.
4. `#./db2setup`.
5. In the Install DB2 V7 screen, highlight the option **DB2 UDB Enterprise Edition** and select it using space bar.
6. Highlight **OK**.
7. Press Enter.
8. On the Create DB2 Services screen, accept the default values: `Do not create a DB2 Instance` and `Do not create the administration server`.
9. Highlight **OK**.
10. Press Enter.
11. A warning `DB2 Instance is not created` appears. Press Enter.
12. A warning: `The Administration Server is not created` appears. Press Enter.
13. A Summary Report appears. Highlight **Continue**.
14. Press Enter.
15. A warning `(X) This is your last chance to stop` appears. Highlight **OK**.

16. Press Enter.
17. At the end of product installation, the screen IBM Product Registration will be showed. Fill it.
18. Click **Next**.
19. Fill in Additional Information.
20. Click **Submit**.
21. Fill the information about the Internet connection.
22. Click **Send**.
23. Press Enter.
24. Press Enter.
25. Press Enter.
26. Highlight **Close**.
27. Press Enter.
28. A warning `DB2 Instance is not created` appears. Highlight **OK**.
29. Press Enter.
30. A warning `The Administration Server is not created` appears. Highlight **OK.**
31. Press Enter.
32. Select **OK** to exit, and press Enter.
33. **`#cd`**
34. **`#unmount /cdrom`**

Repairing the Java path

The installation of DB2 sets the default Java version to 1.1.8. In the file /etc/environment, the Java path is shown in Example 24-4. Note the position of the Java131 directories.

Example 24-4 Java path

```
PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/bin/X11:/sbin:/usr/java131/jre/bin:/
usr/java131/bin
```

To repair it, move the paths to the beginning of the line:

- /usr/java131/jre/bin
- /usr/java131/bin

The corrected file is shown in Example 24-5.

Example 24-5 Corrected file

```
PATH=/usr/java131/jre/bin:/usr/java131/bin:/usr/bin:/etc:/usr/sbin:/usr/ucb:/us
r/bin/X11:/sbin
```

Save the corrected file and log off the system. Log back in and the changes should now take effect. Run the command **`java -fullversion`** to ensure it is now set to the correct level.

24.12 Installing DB2 FixPack 10 for AIX 4.3

For a sample installation of DB2 Fix Pack 10, execute the following steps:

1. Insert the code CD in CD-ROM.
2. #mount /cdrom
3. #cd /cdrom
4. #./installFixPak
5. #cd
6. #unmount /cdrom

24.13 Installing IBM Directory V4.1

For a sample installation of IBM Directory Server, execute the following steps:

1. Insert the code CD in CD-ROM.
2. #mount /cdrom
3. #cd /cdrom
4. #cd ldap41_us
5. #smitty
6. In System Management screen, highlight the option `Software Installation and Maintenance`.
7. Press Enter.
8. In the Software Installation and Maintenance screen, highlight the option **Install and Update Software**.
9. Press Enter.
10. In the Install and Update Software screen, highlight the option **Install and Update from ALL Available Software**.
11. Press Enter.

12. In the field “NPUT device/directory for software on the Install and Update from ALL Available Software screen, insert “.”

13. Press Enter.

14. Highlight the option **Software to Install**.

15. Press F4.

16. In the SOFTWARE to install screen, select the software by pressing F7.

    – ldap.client
    – ldap.html.en_US
    – ldap.msg.en_US
    – ldap.server

17. Press Enter.

18. Press Enter.

19. Press Enter.

20. At the end of the installation process, the screen shown in Example 24-6 will be displayed.

Example 24-6 IDS installation

```
COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

[TOP]
geninstall -I "a -cgNqwX -J"  -Z   -d . -f File 2>&1

File:
    I:ldap.client.adt              4.1.0.0
    I:ldap.client.dmt              4.1.0.0
    I:ldap.client.java             4.1.0.0
    I:ldap.client.rte              4.1.0.0
    I:ldap.html.en_US.config       4.1.0.0
    I:ldap.html.en_US.man          4.1.0.0
    I:ldap.msg.en_US               4.1.0.0
    I:ldap.server.admin            4.1.0.0
    I:ldap.server.cfg              4.1.0.0
[MORE...175]

F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit            /=Find
n=Find Next
```

21. Press F10 to exit.

22. #cd ..
23. #cd gskit
24. #smitty
25. In the System Management screen, highlight the option **Software Installation and Maintenance**.
26. Press Enter.
27. In the Software Installation and Maintenance screen, highlight the option **Install and Update Software**.
28. Press Enter.
29. In the Install and Update Software screen, highlight the option **Install and Update from ALL Available Software**.
30. Press Enter.
31. In the field INPUT device/directory for software on the Install and Update from ALL Available Software screen, insert "**.**"
32. Press Enter.
33. Highlight the option **Software to Install**.
34. Press F4.
35. In the SOFTWARE to install screen, select the software pressing F7:
    - gskkm
36. Press Enter.
37. Press Enter.
38. Press Enter.
39. At the end of the installation process, the screen shown in Example 24-7 appears.

Example 24-7 GSKit installation

```
COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

[TOP]
geninstall -I "a -cgNqwX -J"  -Z   -d . -f File 2>&1

File:
    I:gskkm.rte                   5.0.4.80
```

```
+-----------------------------------------------------------------------------+
                    Pre-installation Verification...
+-----------------------------------------------------------------------------+
Verifying selections...done
Verifying requisites...done
Results...

[MORE...53]

F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit            /=Find
n=Find Next
```

40. Press F10 to exit.
41. #cd
42. #unmount /cdrom

24.14 Installing IBM HTTP Server V1.3.19.3

The IBM HTTP Server will be used for administration of the Directory Server. However, it may also be used for storing pages that can be accessed through Connection Manager.

1. Insert the code CD in CD-ROM.
2. #mount /cdrom
3. #smitty
4. In the System Management screen, highlight the option **Software Installation and Maintenance.**
5. Press Enter.
6. In the Software Installation and Maintenance screen, highlight the option Install and Update Software.
7. Press Enter.
8. In the Install and Update Software screen, highlight the option **Install and Update from ALL Available Software**.
9. Press Enter.
10. In the field INPUT device/directory for software on the Install and Update from ALL Available Software screen, insert "**.**"
11. Press Enter.
12. Highlight the option **Software to Install**.
13. Press F4.

14. In the SOFTWARE to install screen, select the software by pressing F7:
    - http_server.admin
    - http_server.base
    - http_server.html.en_US
    - http_server.man.en_US
    - http_server.modules
    - http_server.msg.en_US.admin
    - http_server.msg.en_US.ssl.core
    - http_server.ssl.128
    - http_server.ssl.core
15. Press Enter.
16. Highlight the field **ACCEPT new license agreements?**
17. Press F4.
18. Highlight **Yes**.
19. Press Enter.
20. Press Enter.
21. Press Enter.
22. At the end of the installation process, the screen shown in Example 24-8 will be displayed.

Example 24-8 IBM HTTP server installation

```
COMMAND STATUS

Command: OK            stdout: yes          stderr: no

Before command completion, additional instructions may appear below.

[TOP]
geninstall -I "a -cgNqwXY -J"  -Z   -d . -f File 2>&1

File:
    I:http_server.admin           1.3.19.3
    I:http_server.base.rte        1.3.19.3
    I:http_server.base.source     1.3.19.3
    I:http_server.html.en_US      1.3.19.3
    I:http_server.man.en_US       1.3.19.3
    I:http_server.modules.fcgi    1.3.19.3
    I:http_server.modules.ldap    1.3.19.3
    I:http_server.modules.ldap.128 1.3.19.3
[MORE...333]

F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit            /=Find
```

```
n=Find Next
```

23. Press F10 to Exit.
24. #cd
25. #unmount /cdrom

24.15 Installation variables

Table 24-1 lists the variables that will be used throughout this section with examples of each.

Table 24-1 Installation variables

Variable	Example
<admin_dn>	cn=root
<admin_dn_pw	password
<o=orgname,c=unitname>	o=ibm,c=us
<virtual_IP_range>	10.0.123.x

24.16 Configuring IBM Directory database

These are the steps:

1. Be sure that IBM HTTP is running. Run the command **`ps -ef|grep http`**. If no HTTPD processes are running, start the IBM HTTP server:
   a. #cd /usr/HTTPServer/bin
   b. #./apachectl start

Example 24-9 HTTP start

```
# ./apachectl start
./apachectl start: httpd started
```

2. Start the LDAP configuration tool.
3. #cd /usr/ldap/bin
4. #./ldapxcfg
5. The window IBM Directory Server Configuration will appear.
6. Select all the options:
   – Set the directory administrator name and password.
   – Create the directory DB2 Database.

   - Configure a Web Server for Directory server administration.

7. Click **Next**.
8. Set the Administrator DN field. (<admin_dn>)
9. Set the administrator password. (<admin_dn_pw>)
10. Confirm the administrator password. (<admin_dn_pw>)
11. Click **Next**.
12. Select **Create a Default LDAPDB2 database**.
13. Click **Next**.
14. Select **Create a Universal DB2 Database (UTF-8).**
15. Click **Next**.
16. At the DB2 Location: field, insert the home directory for the LDAPDB2 instance, such as `/home/ldapdb2`.
17. Click **Next**.
18. Select **IBM HTTP** as the Web server.
19. Click **Next.**
20. In IBM HTTP Web server configuration file, accept the default /usr/HTTPServer/conf/httpd.conf.
21. Clic**k Next**.
22. The Configuration Summary appears.
23. Click **Configure.**
24. The IBM Directory Server configuration results screen appears.
25. Press F4.
26. Restart IBM HTTP:
    a. #cd
    b. #cd /usr/HTTPServer/bin
    c. #./apachectl restart
27. Start the LDAP directory:
    a. #slapd
    b. Check the output of /var/ldap/slapd.err to ensure a successful start.

Enabling auto start of LDAP

The following steps describe the changes required to enable the Directory Server to be started automatically when the system boots:

1. Edit the file /etc/inittab

2. Add the following line to the file:

```
ldapd:2:once:/usr/bin/slapd > /dev/console 2>&1 #autostart LDAP/DB2
Services
```

24.17 Installing Connection Manager and Gatekeeper

These are the steps:

1. Insert the code CD in CD-ROM.
2. #mount /cdrom
3. #cd /cdrom/usr/sys/inst.images
4. #smitty
5. In the System Management screen, highlight the option **Software Installation and Maintenance**.
6. Press Enter.
7. In the Software Installation and Maintenance screen, highlight the option **Install and Update Software**.
8. Press Enter.
9. In the Install and Update Software screen, highlight the option **Install and Update from ALL Available Software**.
10. Press Enter.
11. In the field INPUT device/directory for software on the Install and Update from ALL Available Software screen, insert “**.**”
12. Press Enter.
13. Highlight the option **Software to Install**.
14. Press F4.
15. In the SOFTWARE to install screen, select the software pressing F7:
    - gskkm
    - wg
    - wg.libewgcrypto
    - wg.msg.En_US
    - wg.msg.en_US
    - wgcfg
16. Press Enter.
17. Press Enter.
18. Press Enter.

19. At the end of the installation process, the screen shown in Example 24-10 will appear.

Example 24-10 End of installation process screen

```
COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

[TOP]
geninstall -I "a -cgNqwX -J"  -Z   -d . -f File 2>&1

File:
    I:gskkm.rte                   5.0.5.79
    I:wg.ardis                    5.0.0.0
    I:wg.dataradio                5.0.0.0
    I:wg.datatac                  5.0.0.0
    I:wg.dial                     5.0.0.0
    I:wg.ip-lan                   5.0.0.0
    I:wg.libewgcrypto             1.6.0.0
    I:wg.mobitex                  5.0.0.0
    I:wg.modacom                  5.0.0.0
    I:wg.msg.En_US.rte            5.0.0.0
    I:wg.msg.en_US.rte            5.0.0.0
    I:wg.ppg                      5.0.0.0
    I:wg.rnc3000                  5.0.0.0
[MORE...329]

F1=Help                F2=Refresh             F3=Cancel              F6=Command
F8=Image               F9=Shell               F10=Exit               /=Find
n=Find Next
```

20. Press F10 to exit.
21. #cd
22. #unmount /cdrom

Auto start of Connection Manager

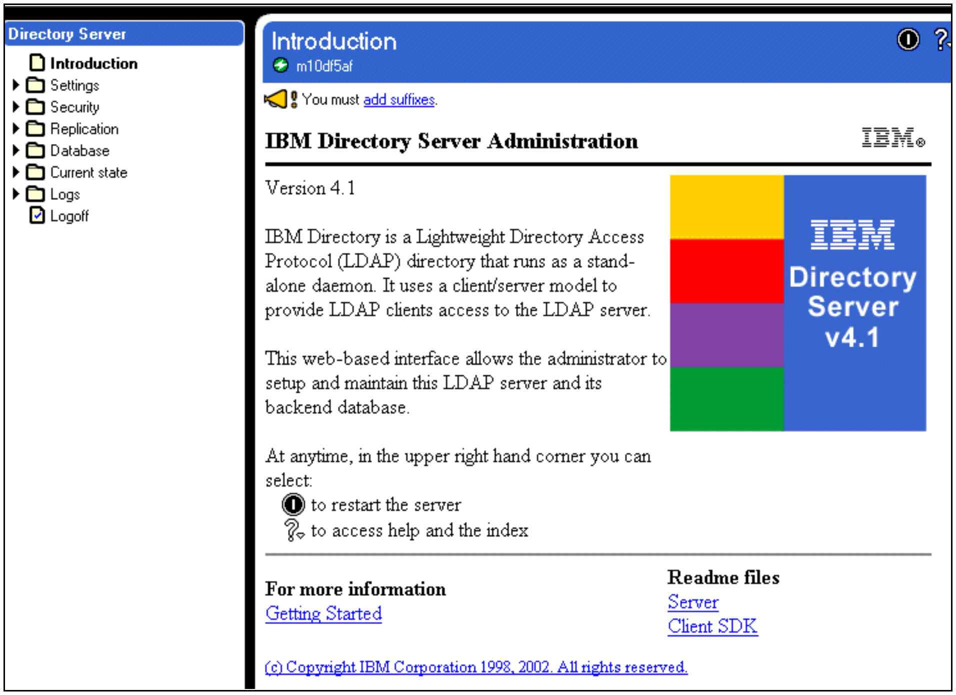
To allow Connection Manager to start automatically at boot time, complete the following steps:

1. Edit the file /etc/inittab
2. If the following line is not present in the file, add it:
   `rcwgated:2:once:/etc/rc.wgated > /dev/console 2>/dev/null`

24.18 Adding suffixes to directory server

These are the steps:

1. Through a Web browser, load the URL `http://<hostname>/ldap`
2. Use <admin_dn> and <admin_dn_pw>.
3. Click **Logon**. A page as shown in Figure 24-4 will be shown.
4. On the left frame select the options **Settings -> Suffixes.**
5. In the Suffix DN field, fill in the Organization Unit as shown in Figure 24-5 using `<o=orgname,c=unitname>`
6. Click **Update**.
7. Click the **Restart Server** button on the top right corner.
8. In the left panel click **Logoff**.
9. In the right panel click **Logoff**.



You must add suffixes.

IBM Directory Server Administration

Version 4.1

IBM Directory is a Lightweight Directory Access Protocol (LDAP) directory that runs as a stand-alone daemon. It uses a client/server model to provide LDAP clients access to the LDAP server.

This web-based interface allows the administrator to setup and maintain this LDAP server and its backend database.

At anytime, in the upper right hand corner you can select:

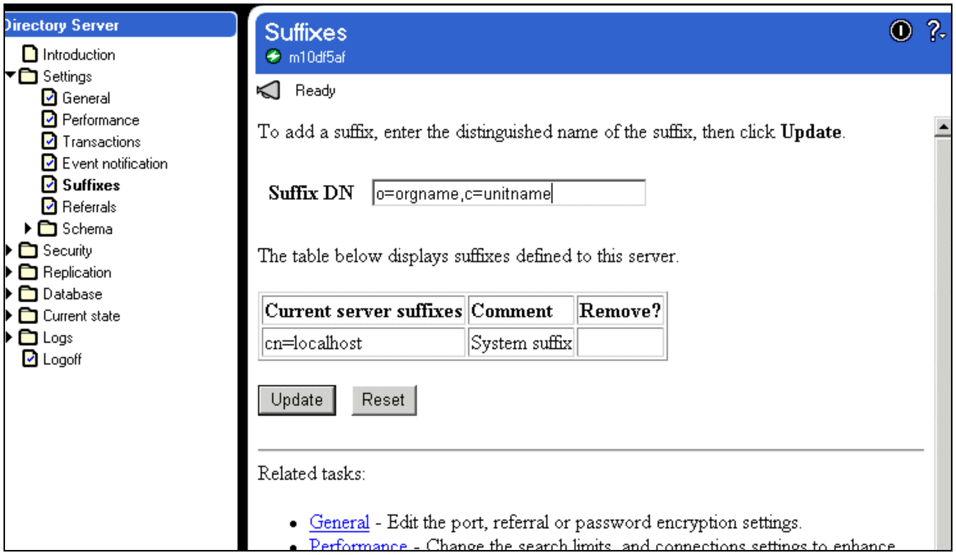
- to restart the server
- to access help and the index

For more information
Getting Started

Readme files
Server
Client SDK

(c) Copyright IBM Corporation 1998, 2002. All rights reserved.

Figure 24-4 IBM Directory Server Web admin view



To add a suffix, enter the distinguished name of the suffix, then click **Update**.

Suffix DN o=orgname,c=unitname

The table below displays suffixes defined to this server.

Current server suffixes	Comment	Remove?
cn=localhost	System suffix	

Related tasks:

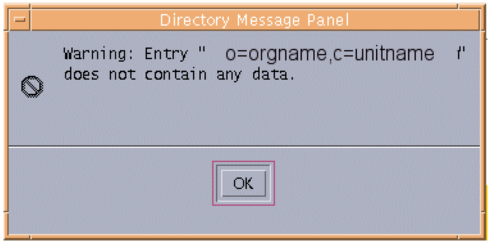
- General - Edit the port, referral or password encryption settings.
- Performance - Change the search limits, and connections settings to enhance

Figure 24-5 IBM Directory Server suffixes

24.19 Configuring organization unit distinguished name

The following steps are required to create an OU through the directory management tool:

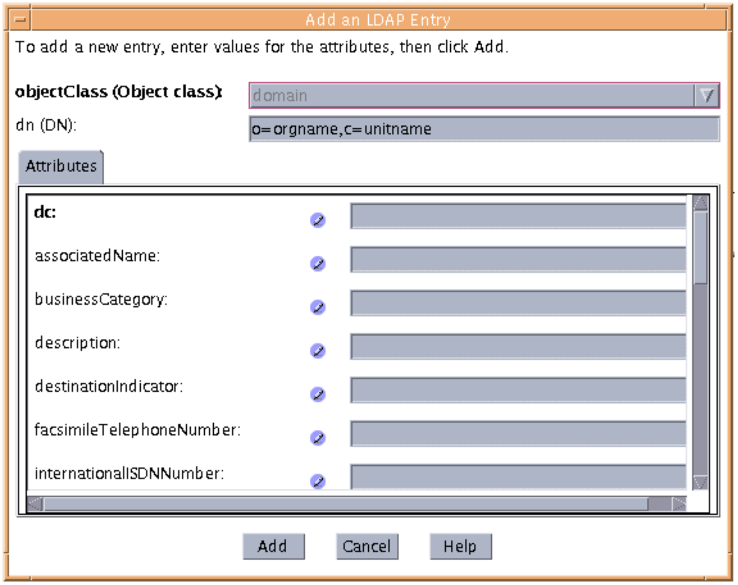
1. #dmt
2. The screen IBM Directory Management Tool appears.
3. In the left frame, select **Rebind.**
4. In the right frame, select the **Authenticated** option.
5. In the User DN field use <admin_dn>.
6. In the User Password field use <admin_dn_pw>.
7. Click **OK.**
8. The warning in Figure 24-6 appears.



```
Warning: Entry "  o=orgname,c=unitname  "
does not contain any data.
```

Figure 24-6 DMT warning

9. Click **OK**.
10. In the right frame, click the **Add** button.
11. In the Entry Type field select **Domain**.
12. In the Parent DN field insert <c=unitname>.
13. In the Entry RDN field insert <o=orgname>.
14. Click **OK**.
15. The screen in Figure 24-7 appears.



To add a new entry, enter values for the attributes, then click Add.

objectClass (Object class): domain

dn (DN): o=orgname,c=unitname

Figure 24-7 Add an LDAP Entry screen

16. In dc field insert <o=orgname,c=unitname>
17. Click **Add**.
18. In left frame press Exit.

24.20 Configuring Gatekeeper

These are the steps:

1. #wgcfg
2. The screen Login profile details appears.
3. Click **Add Profile.**
4. Insert a name for the login Profile Name field.
5. In the Host Name field, insert the host name of the system.
6. Click **OK.**
7. The profile created appears in the Login Profile Details list.

8. Select it and click **OK.**
9. The Gatekeeper Login screen appears.
10. In Login Profile field, select the one that you created.
11. In the Organization Unit field, use <o=orgname,c=unitname>.
12. In the Administrator ID field, use system root user.
13. In Password: field, use password for the system root user.
14. Click **Login.**
15. Read the License Agreement. If you agree, click **Accept.**
16. Close the Help window by pressing Alt+F4.
17. The Configure the Gatekeeper Access Manager screen appears.
18. Click **Next.**
19. In the Administrator's Distinguished Name (DN) field use <admin_dn>.
20. In Administrator's Password field use <admin_pw>.
21. In Primary DSS IP address field, use the IP of the system.
22. In the other fields keep default.
23. Click **Next.**
24. In the Base Distinguish Name (DN): field, use <o=orgname,c=unitname>
25. Click **Next**.
26. Keep the default.
27. Click **Next.**
28. Select the log level required.
29. Click **Finish**. This will take some time as the LDAP server is updated with the Connection Manager configuration items.
30. When finished, the screen shown in Figure 24-8 on page 548 appears.
31. In the left frame select the **Resources** tab.
32. Right-click the folder named **WECM.**
33. Select **Add Resource.**
34. Select **Connection Manager.**
35. The Gatekeeper Login screen appears.
36. Insert the system root password in the Password field.
37. Click **Login.**
38. In Gateway Identifier insert some identifier name or accept the default.
39. Click **Next.**
40. Click **Next.**
41. In the Connection Manager DB2 client instance ID field, accept the default.
42. In the Home directory for DB2 client instance ID field, accept the default.
43. Click **Next**.

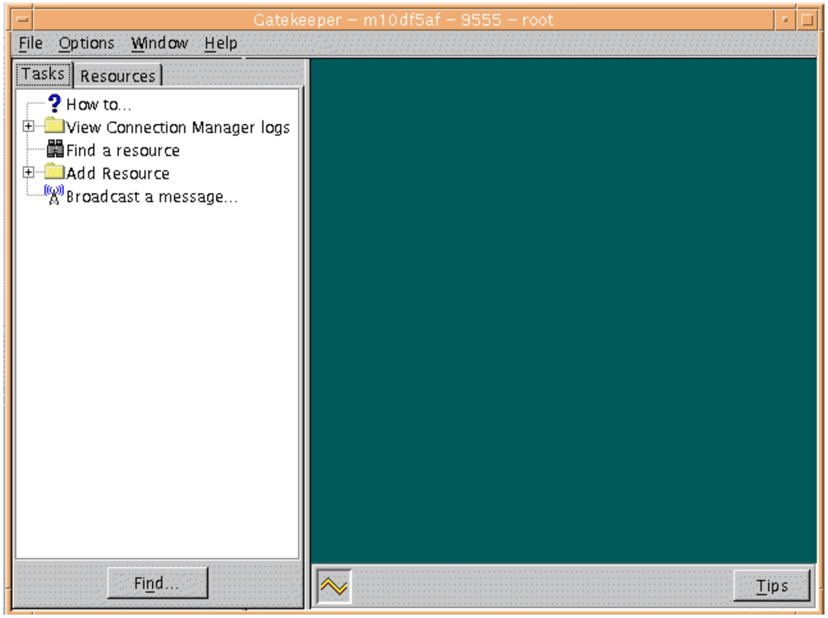
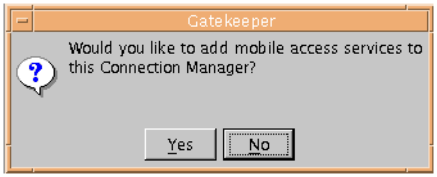


Figure 24-8 Gatekeeper View

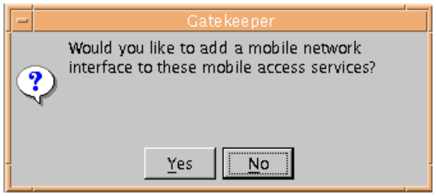
44. Accept the default.
45. Click **Next**.
46. Accept the default.
47. Click **Next**.
48. Accept the default.
49. Click **Next**.
50. Accept the default.
51. Click **Next**.
52. Accept the default.
53. Click **Next**.
54. Accept the default.
55. Click **Next**.
56. Click **Finish**.
57. A warning screen appears as in Figure 24-9.



Would you like to add mobile access services to this Connection Manager?

Figure 24-9 Warning - Mobile Access Services

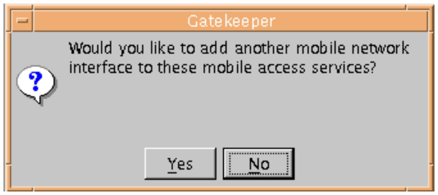
58. Click **Yes.**
59. Accept the default.
60. Click **Finish**.
61. A warning appears as in Figure 24-10.


Gatekeeper

Would you like to add a mobile network interface to these mobile access services?

Figure 24-10 Warning - Mobile Network Interface

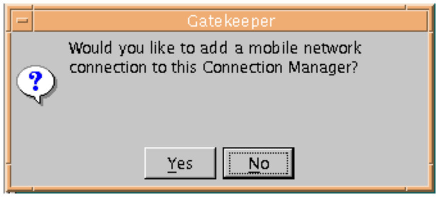
62. Click **Yes**.
63. Accept the default.
64. Click **Next**.
65. Insert the first address of the range designated to the virtual IPs (<virtual_IP_range> where x=1).
66. Click **Next**.
67. Accept the default.
68. Click **Next.**
69. Accept the default.
70. Click **Next.**
71. Accept the default.
72. Click **Finish.**
73. A warning appears as in Figure 24-11.


Gatekeeper

Would you like to add another mobile network interface to these mobile access services?

Figure 24-11 Warning - Another Mobile Network Interface

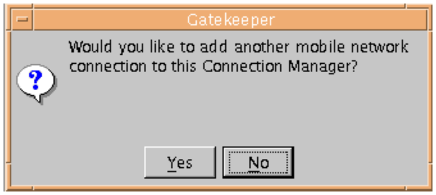
74. Click **No**.
75. A message appears relating to the WAP proxy. The WAP configuration will not be described in this chapter.
76. Click **No**.
77. Click **No** to indicate messaging services are not required.
78. A warning appears as in Figure 24-12.


Gatekeeper

Would you like to add a mobile network connection to this Connection Manager?

Figure 24-12 Warning - Mobile Network Connection

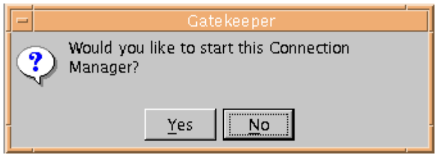
79. Click **Yes**.
80. Select the **ip-lan IP LAN-based networks** option.
81. Click **OK**.
82. Accept the default.
83. Click **Next**.
84. Accept the default.
85. Click **Finish**.
86. A warning appears as in Figure 24-13.


Gatekeeper

Would you like to add another mobile network connection to this Connection Manager?

Figure 24-13 Warning - Another Mobile Network Connection

87. Click **No**.
88. A warning appears as in Figure 24-14.


Gatekeeper

Would you like to start this Connection Manager?

Figure 24-14 Warning - Start Connection Manager

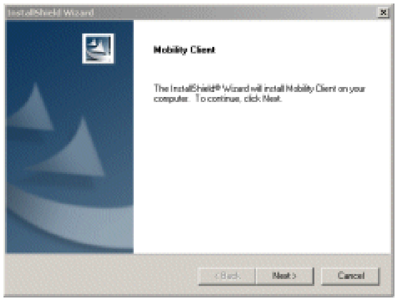
89. Click **Yes**.
90. In the left frame, expand the **Default Resources** folder.
91. Expand the created **Connection Manager** folder.
92. Right-click the **Default Resources** folder.
93. Select **Add Resource.**
94. Select **User.**
95. In User ID field insert a user name.
96. In the Full Name field insert the full name of the created user.

97. Click **Next.**
98. Accept the default.
99. Click **Next**.
100. Set the password to the user created.
101. Confirm the password.
102. Accept the default.
103. Click **Next.**
104. Accept the default.
105. Click **Next.**
106. Accept the default.
107. Click **Next**.
108. A warning appears (`Related to Mobile Device`).
109. Click **No.**
110. In menu at the top, click **File.**
111. Click **Exit**.

24.21 Installing Mobility Client in a Windows system

These are the steps:

1. Insert the code CD in CD-ROM.
2. Click the **Start** button.
3. Select the option **Run**.
4. In the open field insert: `<CDROM_Drive_Letter>\clients\Win32\WC_Win32.exe`
5. Click **OK.**
6. The InstallShield wizard appears as in Figure 24-15.



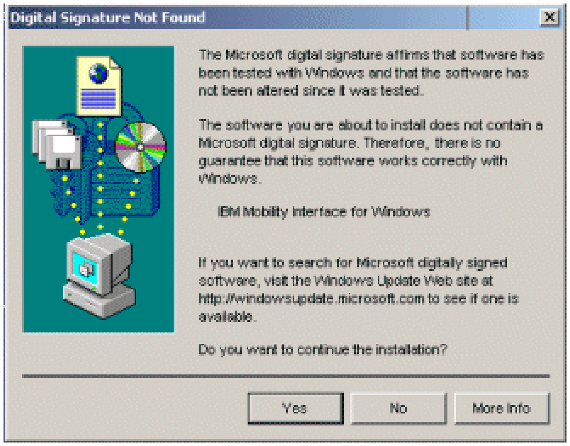
Mobility Client

The InstallShield® Wizard will install Mobility Client on your computer. To continue, click Next.

Figure 24-15 InstallShield Wizard

7. Click **Next.**
8. Click **Next**.
9. Accept the default.
10. Click **Next**.
11. Click **Typical.**

12. Click **Next**.
13. Click **Next.** Wait while the installation completes.
14. A warning appears as in Figure 24-16.



Digital Signature Not Found

The Microsoft digital signature affirms that software has been tested with Windows and that the software has not been altered since it was tested.

The software you are about to install does not contain a Microsoft digital signature. Therefore, there is no guarantee that this software works correctly with Windows.

IBM Mobility Interface for Windows

If you want to search for Microsoft digitally signed software, visit the Windows Update Web site at http://windowsupdate.microsoft.com to see if one is available.

Do you want to continue the installation?

Figure 24-16 Warning - Certificate

15. Click **Yes.**
16. Click **Finish**.
17. The wireless client window appears as in Figure 24-17.

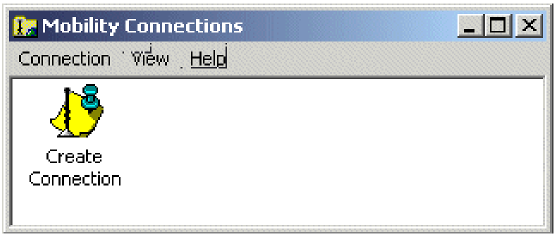
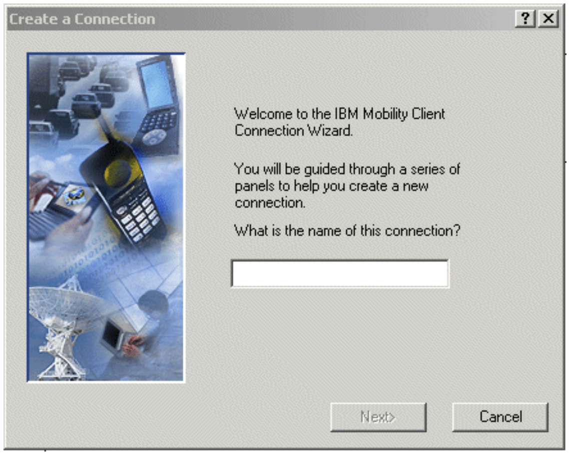


Figure 24-17 Wireless Client - Mobility Connections window

18. Double-click the **Create Connection** icon.
19. The Create Connection wizard appears as in Figure 24-18.



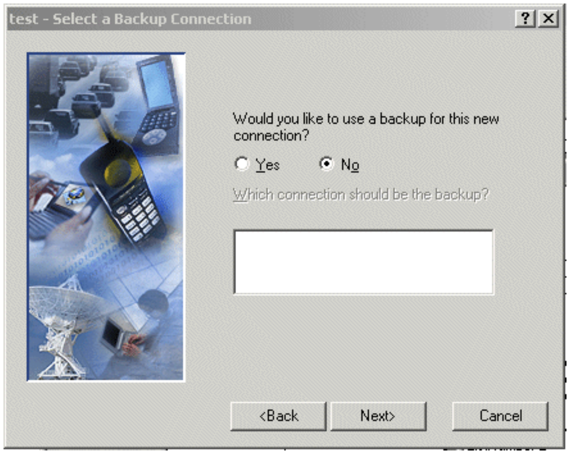
Welcome to the IBM Mobility Client Connection Wizard.

You will be guided through a series of panels to help you create a new connection.

What is the name of this connection?

Figure 24-18 Wireless Client Wizard - Create connection

20. Give a name for the connection.
21. Click **Next**.
22. The Select Backup Connection appears as in Figure 24-19.



Would you like to use a backup for this new connection?

○ Yes ● No

Which connection should be the backup?

Figure 24-19 Wireless Client Wizard - Select Backup Connection screen

23. Select **No**.
24. Click **Next**.
25. The Select Network screen appears as in Figure 24-20.

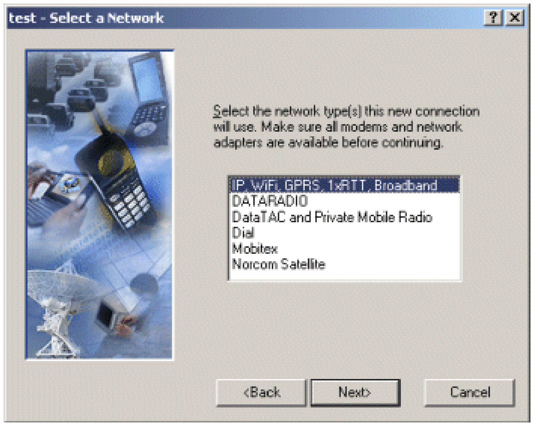

test - Select a Network
? X
Select the network type(s) this new connection will use. Make sure all modems and network adapters are available before continuing.
IP, WiFi, GPRS, 1xRTT, Broadband
DATARADIO
DataTAC and Private Mobile Radio
Dial
Mobitex
Norcom Satellite
<Back
Next>
Cancel

Figure 24-20 Wireless Client Wizard - Select Network screen

26. Select the **IP, WiFi, GPRS, 1xRTT, Broadband** option.
27. Click **Next**.
28. The Network Setup - IP Based screen appears as in Figure 24-21.

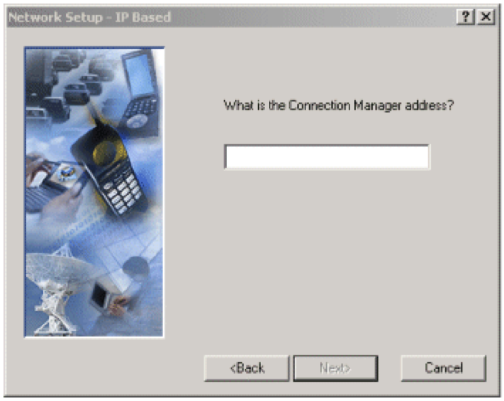
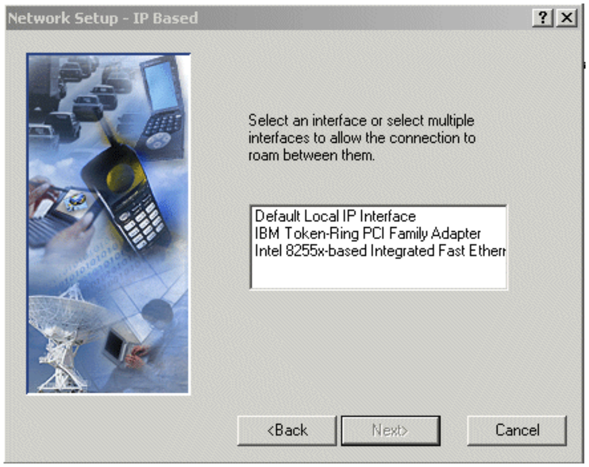


Figure 24-21 Wireless Client Wizard - Network Setup - IP based

29. Set the IP of WebSphere Everyplace Connection Manager host system.
30. Click **Next**.
31. The Select Interface... screen appears as in Figure 24-22.



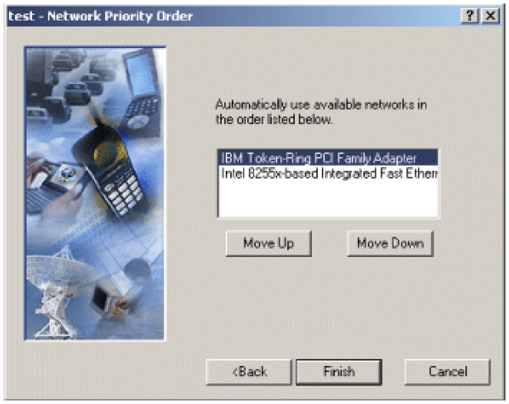
Network Setup - IP Based

Select an interface or select multiple interfaces to allow the connection to roam between them.

Default Local IP Interface
IBM Token-Ring PCI Family Adapter
Intel 8255x-based Integrated Fast Ethern

Figure 24-22   Wireless Client Wizard - select interface

32. Select the interfaces wanted.
33. Click **Next.**
34. Click **Next**.
35. If multiple interfaces were selected, the Network Priority Order screen will appear as in Figure 24-23.



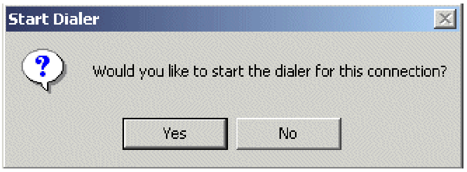
test - Network Priority Order

Automatically use available networks in the order listed below.

IBM Token-Ring PCI Family Adapter
Intel 8255x-based Integrated Fast Ethern

Figure 24-23   Wireless Client Wizard - Network Priority Order

36. Move up and down the interfaces to select the priority.
37. Click **Finish**.
38. The Start Dialer screen appears as in Figure 24-24.



Would you like to start the dialer for this connection?

Figure 24-24 Warning - Start Dialer

39. Click **Yes**.
40. The Dialer appears as in Figure 24-25.

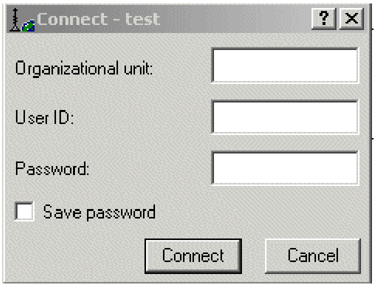


Figure 24-25 Wireless client dialer

41. In the Organization Unit field insert `<o=orgname,c=unitname>`
42. In the User ID field insert the user name.
43. In the Password field insert the password of this user.
44. If wanted, select the option **Save Password**.
45. Click **Connect**.
46. The Wireless Client icon will appear in the task bar as in Figure 24-26.

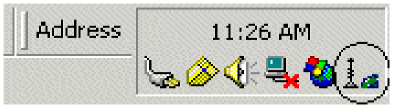


Figure 24-26 Wireless Client Icon

The system is ready. For further details on configuring the client, refer to Chapter 10., “Mobility Clients” on page 171.

B

Simple device resolver implementation using NAS and RADIUS

This chapter describes how to set up an environment that uses a network access server for dial-in access to the network; a RADIUS server for authentication; and the device resolver in Connection Manager for user identification.

This chapter describes following topics:

- Network Access Server configuration
- FreeRADIUS installation and configuration
- Configure device resolver
- Testing the configuration

24.22 Overview

This environment is meant to be a simple example on how to integrate NAS and Connection Manager authentication using RADIUS.

Our sample environment is a typical WAP service provider environment where users connect to the network using a dial-up connection. DHCP is used to assign a dynamic IP address to a client device. A user ID and password are required to access the WAP gateway. See Figure 24-27.

The sample environment consists of the following components:

- Network Access Server (NAS)
- RADIUS server
- WebSphere Everyplace Connection Manager (WECM)
- LDAP server
- Dial-up client

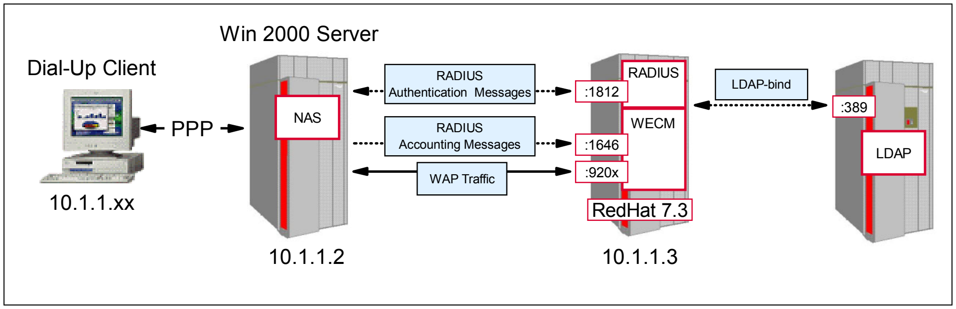


Figure 24-27 Sample environment

Network Access Server

The NAS provides dial-up services for the client devices. In our environment, the NAS is a Windows 2000 Server with Remote Access Services (RAS).

When client calls to the NAS, it checks user credentials from the RADIUS server by sending an *Access-Request* RADIUS message. If the authentication is successful, NAS sends an *Accounting-Start* RADIUS message to the Connection Manager, so Connection Manager can recognize the user by the IP address of incoming traffic. When the dial-up client disconnects, NAS sends an *Accounting-Stop* RADIUS message to the Connection Manager, so Connection Manager knows that the user is offline, and that traffic from the given IP is not from the user anymore.

RADIUS server

The Remote Authentication Dial-In User Service (RADIUS) server provides an remote authentication service for Network Access Servers (NAS). In our environment, the RADIUS server is the FreeRADIUS RADIUS server running on a Linux box (Red Hat 7.3).

When RADIUS receives an `Access-Request` message from the NAS, it checks the user credentials using the configured authentication method (in our environment, the local OS authentication). If the credentials were correct, an `Access-Accept` message is sent back to NAS.

Connection Manager

Connection Manager provides a WAP gateway and proxy services for WAP clients. In our environment, the Connection Manager is running on a Linux box (Red Hat 7.3).

When Connection Manager receives an `Accounting-Start` message from NAS, it gets the username and IP address from the message and maps them. So, whenever there is traffic from that IP address, Connection Manager trusts that it is from that user. When Connection Manager receives an `Accounting-Stop` message from NAS, it brakes the mapping, and therefore stops trusting the IP address.

Dial-up client

The dial-up client is usually a WAP phone, which connects to the network using GSM-data by dialing to the service providers modem. In our environment, the client is a laptop with a WAP emulator connected to the NAS with a null-modem cable.

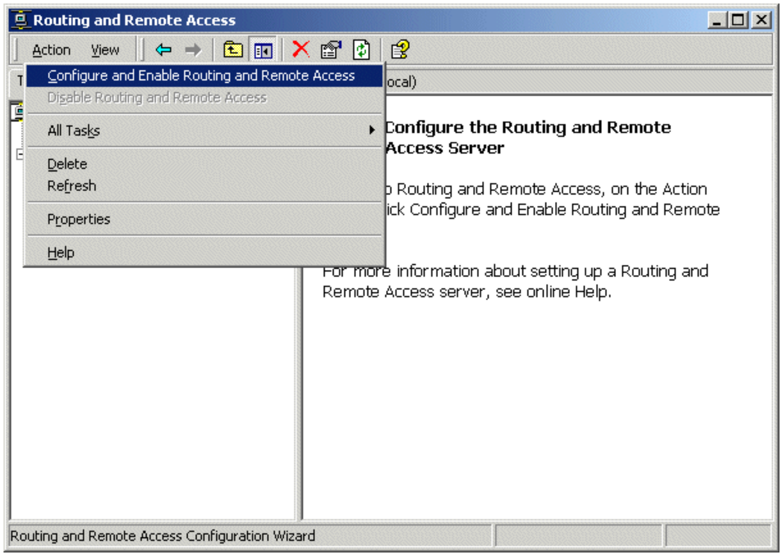
24.23 Network Access Server configuration

The prerequisites to configure Network Access Server are:

- Windows 2000 Server installed and configured
- Routing and Remote Access Server installed
- Modem (or null-modem cable) configured for dial-in access
- Network and DHCP configured

24.23.1 Configuration

First, open the **Routing and Remote Access** from Administrative Tools, then select the RAS server from the list, right-click and select **Configure and Enable Routing and Remote Access** (Figure 24-28).


Routing and Remote Access
Action
View
Configure and Enable Routing and Remote Access
Disable Routing and Remote Access
All Tasks
Delete
Refresh
Properties
Help
ocal)
Routing and Remote Access Configuration Wizard

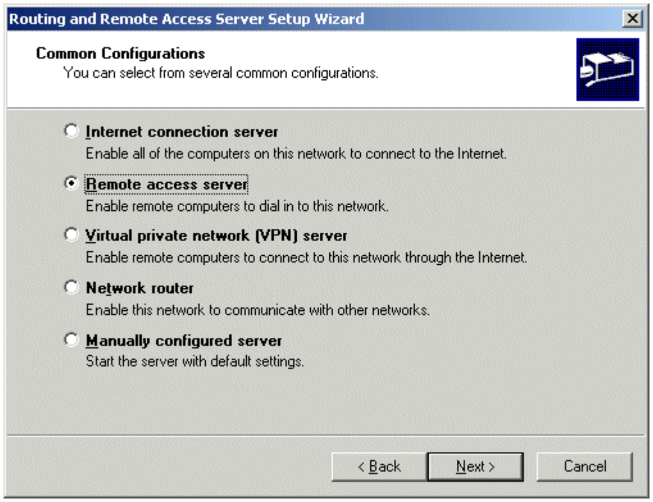
Configure the Routing and Remote Access Server

o Routing and Remote Access, on the Action ick Configure and Enable Routing and Remote

For more information about setting up a Routing and Remote Access server, see online Help.

Figure 24-28 Configure and Enable Routing and Remote Access

When the configuration wizard opens, select **Next** to close the welcome page. From Common Configurations, select **Remote access server** and click **Next** (Figure 24-29).



Common Configurations

You can select from several common configurations.

○ **Internet connection server**

Enable all of the computers on this network to connect to the Internet.

● **Remote access server**

Enable remote computers to dial in to this network.

○ **Virtual private network (VPN) server**

Enable remote computers to connect to this network through the Internet.

○ **Network router**

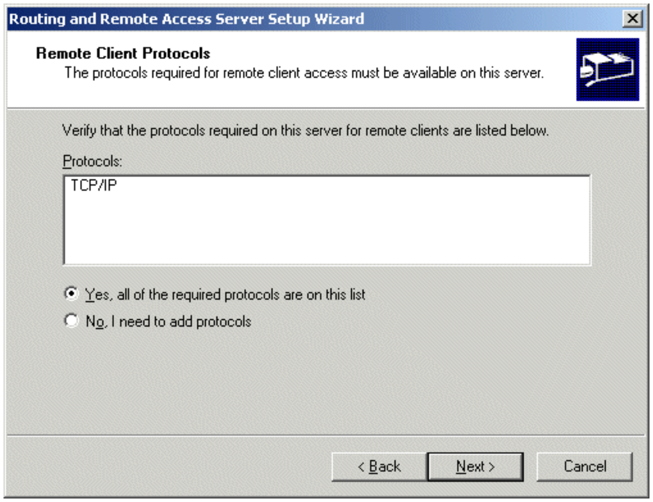
Enable this network to communicate with other networks.

○ **Manually configured server**

Start the server with default settings.

Figure 24-29 Common Configurations, Remote access server

On Remote Client Protocols, check that at least the TCP/IP protocol is on the list (Figure 24-30). If it is, select option **Yes, all required protocols are on this list** and click **Next**. If not, select option **No, I need to add protocol** and click **Next** to install required protocols.



Routing and Remote Access Server Setup Wizard

Remote Client Protocols
The protocols required for remote client access must be available on this server.

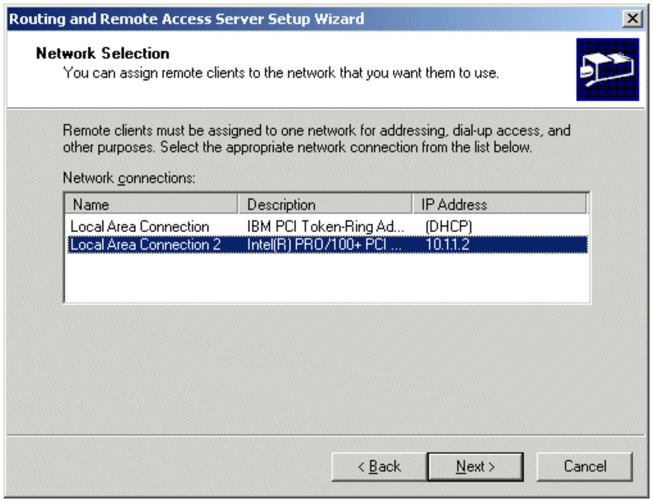
Verify that the protocols required on this server for remote clients are listed below.

Protocols:
TCP/IP

● Yes, all of the required protocols are on this list
○ No, I need to add protocols

Figure 24-30 Remote Client Protocols

From Network selection, select the network interface that you are using to connect to your network and click **Next**.



Routing and Remote Access Server Setup Wizard

Network Selection
You can assign remote clients to the network that you want them to use.

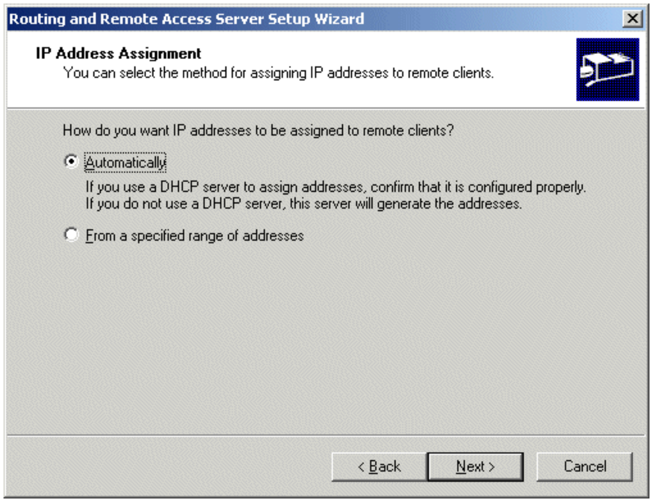
Remote clients must be assigned to one network for addressing, dial-up access, and other purposes. Select the appropriate network connection from the list below.

Network connections:

Name	Description	IP Address
Local Area Connection	IBM PCI Token-Ring Ad...	(DHCP)
Local Area Connection 2	Intel(R) PRO/100+ PCI ...	10.1.1.2

Figure 24-31 Network selection

From IP Address Assignment, select **Automatically** and click **Next** (Figure 24-32).


Routing and Remote Access Server Setup Wizard

IP Address Assignment
You can select the method for assigning IP addresses to remote clients.

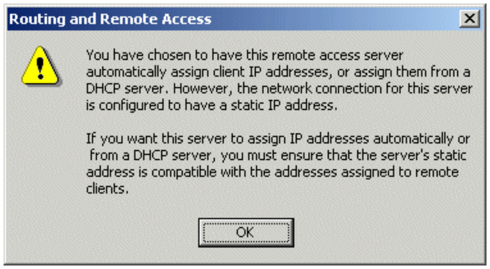
How do you want IP addresses to be assigned to remote clients?

● Automatically
If you use a DHCP server to assign addresses, confirm that it is configured properly.
If you do not use a DHCP server, this server will generate the addresses.

○ From a specified range of addresses

Figure 24-32 IP Address Assignment

If the network interface you select is configured to use a static IP address, a warning will be displayed (Figure 24-33). This simply means that all addresses, DHCP, and static must be in same subnet. Click **OK**.

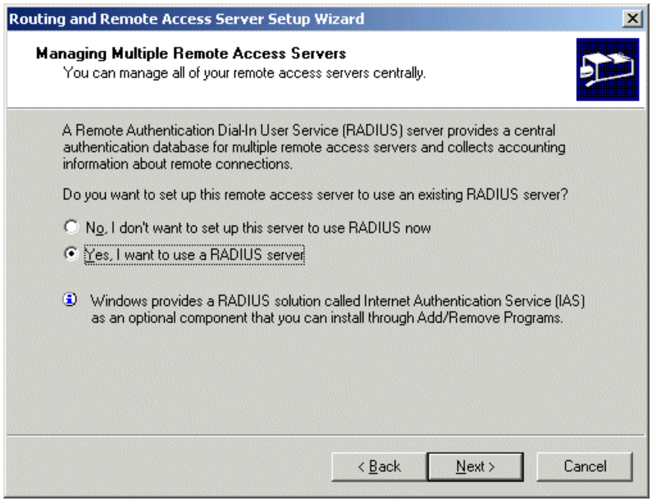

Routing and Remote Access

You have chosen to have this remote access server automatically assign client IP addresses, or assign them from a DHCP server. However, the network connection for this server is configured to have a static IP address.

If you want this server to assign IP addresses automatically or from a DHCP server, you must ensure that the server's static address is compatible with the addresses assigned to remote clients.

Figure 24-33 DHCP Warning

From Managing Multiple Remote Access Servers, select **Yes, I want to use a RADIUS server** and click **Next** (Figure 24-34).


Routing and Remote Access Server Setup Wizard

Managing Multiple Remote Access Servers

You can manage all of your remote access servers centrally.

A Remote Authentication Dial-In User Service (RADIUS) server provides a central authentication database for multiple remote access servers and collects accounting information about remote connections.

Do you want to set up this remote access server to use an existing RADIUS server?

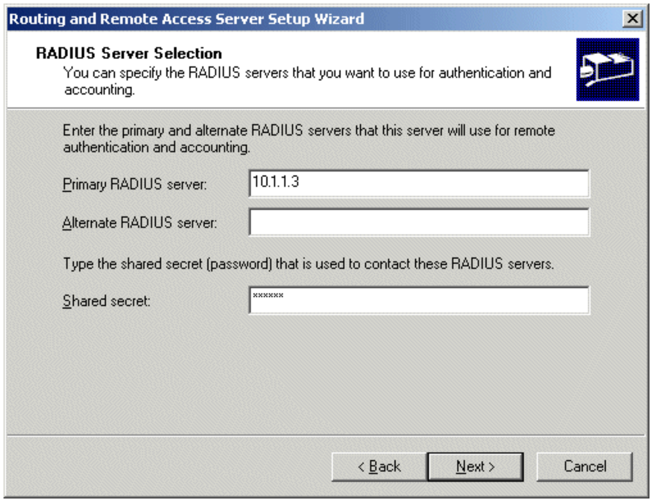
○ No, I don't want to set up this server to use RADIUS now

● Yes, I want to use a RADIUS server

Windows provides a RADIUS solution called Internet Authentication Service (IAS) as an optional component that you can install through Add/Remove Programs.

Figure 24-34 Managing Multiple Remote Access Servers

In RADIUS Server Selection, enter the IP address of your RADIUS server. Enter is also a shared secret, which is used to encrypt the RADIUS messages (Figure 24-35). Click **Next**.


Routing and Remote Access Server Setup Wizard

RADIUS Server Selection

You can specify the RADIUS servers that you want to use for authentication and accounting.

Enter the primary and alternate RADIUS servers that this server will use for remote authentication and accounting.

Primary RADIUS server: 10.1.1.3

Alternate RADIUS server:

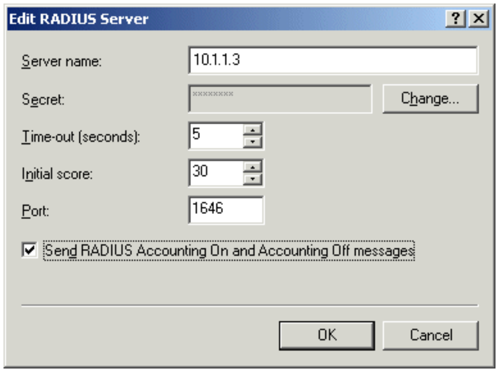
Type the shared secret (password) that is used to contact these RADIUS servers.

Shared secret: ******

Figure 24-35 RADIUS Server Selection

From the next page, click **Finish** to close the setup wizard. The DHCP relay warning window may pop-up, but you can ignore that.

Next, select the **RAS server** from the list again, right-click and select **Properties**. On Security tab click the **Edit** button next to RADIUS accounting. Enter the server name or IP address of the Connection Manager server. Change the default port from 1813 to 1646, which is the default for RADIUS accounting in Connection Manager. Then check the **Send RADIUS Accounting On and Accounting Off** messages to enable accounting messages. If you are using a different secret between NAS and Connection Manager, then between NAS and RADIUS, change the secret by clicking the **Change...** button. Click **OK** to apply the settings (Figure 24-36).



Edit RADIUS Server

Server name: 10.1.1.3

Secret: xxxxxxxx [Change...]

Time-out (seconds): 5

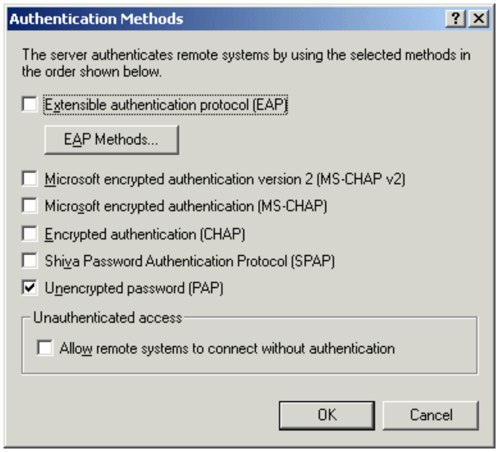
Initial score: 30

Port: 1646

☑ Send RADIUS Accounting On and Accounting Off messages

Figure 24-36 Accounting settings

Next, on the Authentication tab, click **Authentication methods**. In our environment, we used only unencrypted passwords, PAP. Uncheck all other options (Figure 24-37) and click **OK** to apply.



Authentication Methods

The server authenticates remote systems by using the selected methods in the order shown below.

☐ Extensible authentication protocol (EAP)

☐ Microsoft encrypted authentication version 2 (MS-CHAP v2)

☐ Microsoft encrypted authentication (MS-CHAP)

☐ Encrypted authentication (CHAP)

☐ Shiva Password Authentication Protocol (SPAP)

☑ Unencrypted password (PAP)

Unauthenticated access

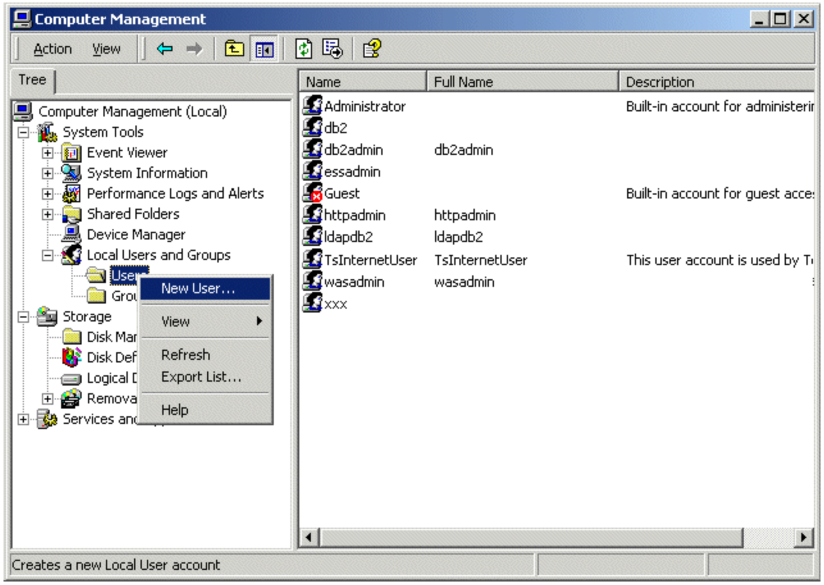
☐ Allow remote systems to connect without authentication

Figure 24-37 Authentication methods

Note: You have to restart routing and remote access to changes to take effect.

24.23.2 Creating testuser account

Launch **Computer Management** tool and select the **Users** folder from Local Users and Groups. Right-click and select **New User...** (Figure 24-38).



Name	Full Name	Description
Administrator		Built-in account for administerir
db2		
db2admin	db2admin	
essadmin		
Guest		Built-in account for guest acce:
httpadmin	httpadmin	
ldapdb2	ldapdb2	
TsInternetUser	TsInternetUser	This user account is used by T
wasadmin	wasadmin	
xxx		

Creates a new Local User account

Figure 24-38 Create new user

Enter `testuser` as the user name, and `password` as a password. Uncheck all check boxes and click **OK**.

Note: testuser needs to have the right to dial-in; check the setting from the user properties.

24.24 FreeRADIUS installation and configuration

FreeRADIUS runs on Linux, FreeBSD, OpenBSD, and Solaris. For the sake of simplicity, the RADIUS server is installed on the same server as WebSphere Everyplace Connection Manager, which is Red Hat Linux 7.3.

FreeRADIUS is only available as a source package and must be compiled on the target system. If LDAP-bind will be used as the authentication mechanism in RADIUS, then the OpenLDAP development package must be installed.

24.24.1 Installation

The FreeRADIUS server is available for download at:
http://www.freeradius.org

It is only provided as a source package, either as a source tarball or as an extract from CVS. Version 0.9.0 of FreeRADIUS, which is the latest version as of writing this book, has been used for the sample scenario.

In order to use a LDAP server as the authentication source, FreeRADIUS must be compiled with LDAP support. To do so, the OpenLDAP development package (see http://www.openldap.org) must be installed when compiling the package.

1. Log in as a root user and go to the directory where you downloaded `freeradius.tar.gz`
2. Extract the contents of the archive. It will extract all files to a subdirectory called `freeradius-x.x.x` where `x.x.x` is the version of FreeRADIUS.

   ```
   # tar -zxvf freeradius.tar.gz
   ```
3. Change to the freeradius directory:

   ```
   # cd freeradius-0.9.0
   ```
4. FreeRADIUS has autoconf support. If you do not need to modify the default configuration, run the following command to configure FreeRADIUS:

   ```
   # ./configure
   ```

   FreeRADIUS will be installed in the directory /usr/local. If a different installation directory should be required, run the following command:

   ```
   # ./configure --prefix=<install_dir>
   ```

   where <*install_dir*> is installation directory. To list all available options, run the following command:

   ```
   # ./configure --help
   ```
5. Compile FreeRADIUS:

   ```
   # make
   ```
6. Finally, install all components:

   ```
   # make install
   ```

24.24.2 Configuration

All configuration of FreeRADIUS is done through text files. These files can be found in the directory /usr/local/etc/raddb.

First, the network access server must be added as a trusted client so that the RADIUS server accepts messages from NAS:

1. Go to FreeRADIUS configuration directory:

   ```
   # cd /usr/local/etc/raddb
   ```

2. Open the `clients.conf` file for editing:

   ```
   # vi clients.conf
   ```

3. Add the following lines to the end of the file:

   ```
   client <nas_ip> {
         secret    = <secret>
         shortname = <shortname>
   }
   ```

Replace *<nas_ip>* with the IP address of the NAS server, and *<secret>* with the secret you provided during the NAS configuration. A shortname can be given to this client entry by replacing *<shortname>*.

This client entry tells FreeRADIUS to trust RADIUS messages from NAS.

> **Note:** If NAS IP or secret are wrong, all RADIUS messages from NAS are rejected.

FreeRADIUS provides various options to authenticate users. See the documentation for all supported options. For the sample scenario, two types of authentication have been used:

- User credentials in a file
- LDAP-bind authentication against Connection Manager LDAP

User credentials in a file

All authentication configuration is done in the file `users`, which can be found in the directory /usr/local/etc/raddb:

1. Open the `users` file for editing:

   ```
   # vi users
   ```

2. Add following line to the end of the file and save:

   ```
   testuser        Auth-Type := Local, User-Password == "password"
   ```

   This will create a user with the user ID `testuser` and the password `password`.

Now, FreeRADIUS is configured to check testuser credentials from the users file.

LDAP-bind authentication

The LDAP authentication module is not enabled in the default installation. In order to use the LDAP-bind authentication, the appropriate module in FreeRADIUS must be enabled, the LDAP server must be configured, and the authentication mechanism must be set to LDAP:

1. Enable LDAP authentication module in the file radiusd.conf, which is located in /usr/local/etc/raddb. Open the file for editing, locate the authenticate section and uncomment the Auth-Type LDAP section.

   ```
   authenticate {
   ...
      Auth-Type LDAP {
         ldap
      }
   }
   ```

2. Configure the LDAP server in radiusd.conf. Open the file for editing and locate the modules section. The LDAP server configuration is done in the ldap section within the modules section.

   ```
   modules {
      ...
      ldap {
         server = “<ldap_server>“
         basedn = “<base_dn>“
         filter = “(uid=%{Stripped-User-Name:-%{User-Name}})“
         ...
      }
      ...
   }
   ```

   where <ldap_server> is the hostname or IP address of the Connection Manager LDAP server, and <base_dn> is the base distinguished name of the users. All other properties can be left at the default values.

3. Set the authentication mechanism to LDAP-bind in the users file. Locate the following section in the file:

   ```
   DEFAULT Auth-Type = System
      Fall-Through = 1
   ```

   and change System to LDAP. The section should look like this:

   ```
   DEFAULT Auth-Type = LDAP
      Fall-Through = 1
   ```

FreeRADIUS is now configured to use LDAP-bind authentication.

24.24.3 Starting FreeRADIUS

Debug mode

To start FreeRADIUS in debug mode, type the following command:

```
# radiusd -X
```

This starts up FreeRADIUS, and all messages are dumped to console. The last lines in the startup dump should be similar to:

```
Listening on IP address *, ports 1812/udp and 1813/udp, with proxy on 1814/udp.
Ready to process requests.
```

Normal mode

To start FreeRADIUS in normal mode, just type:

```
# radiusd
```

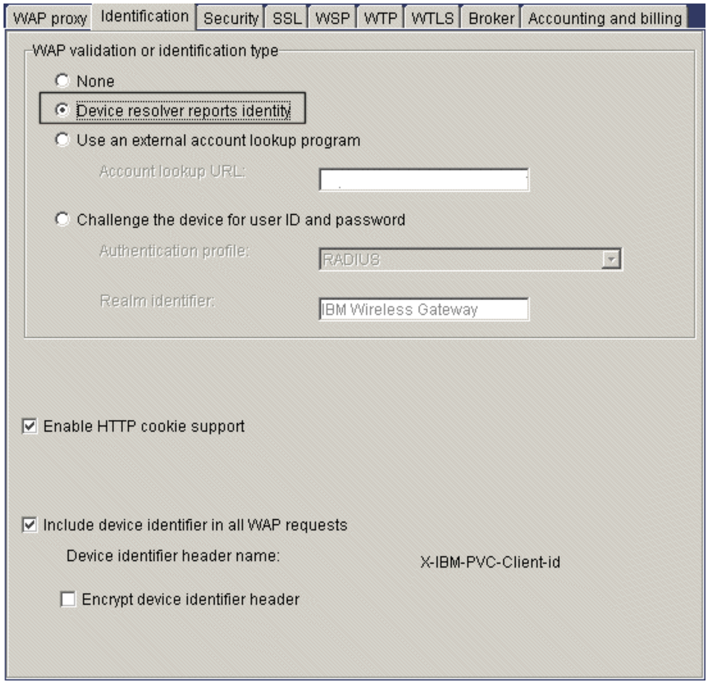
The log files are located in `/usr/local/var/log/radius`

24.25 Configure device resolver

It assumed that Connection Manager is already installed and configured.

24.25.1 WAP proxy configuration

Open **WAP proxy properties** and go to the **Identification** tab. Check the option **Device resolver reports identity** (Figure 24-39).



WAP validation or identification type

○ None

● Device resolver reports identity

○ Use an external account lookup program

Account lookup URL:

○ Challenge the device for user ID and password

Authentication profile: RADIUS

Realm identifier: IBM Wireless Gateway

☑ Enable HTTP cookie support

☑ Include device identifier in all WAP requests

Device identifier header name: X-IBM-PVC-Client-id

□ Encrypt device identifier header

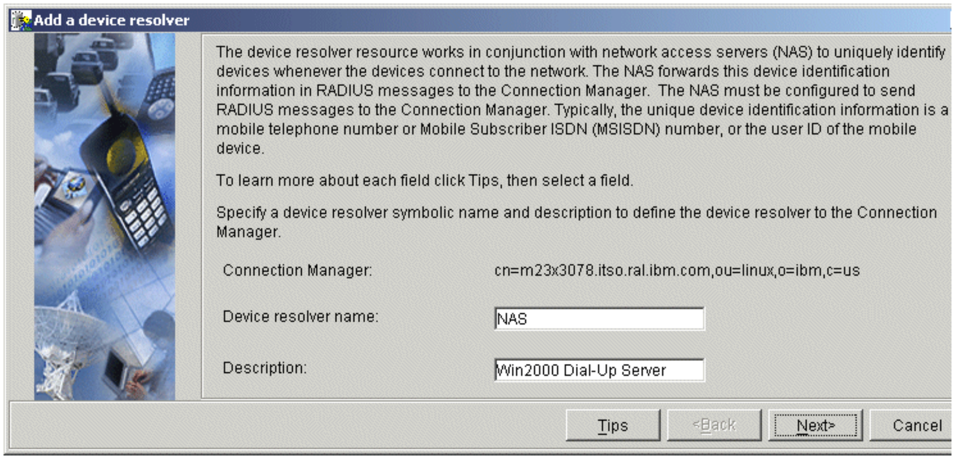
Figure 24-39 WAP Proxy configuration

Click **OK**.

24.25.2 Device resolver installation

From Gatekeeper, select your Connection Manager server from the list, right-click and select **Add ->Device resolver**.

Enter a name and description for the device resolver and click **Next.**


Add a device resolver

The device resolver resource works in conjunction with network access servers (NAS) to uniquely identify devices whenever the devices connect to the network. The NAS forwards this device identification information in RADIUS messages to the Connection Manager. The NAS must be configured to send RADIUS messages to the Connection Manager. Typically, the unique device identification information is a mobile telephone number or Mobile Subscriber ISDN (MSISDN) number, or the user ID of the mobile device.

To learn more about each field click Tips, then select a field.

Specify a device resolver symbolic name and description to define the device resolver to the Connection Manager.

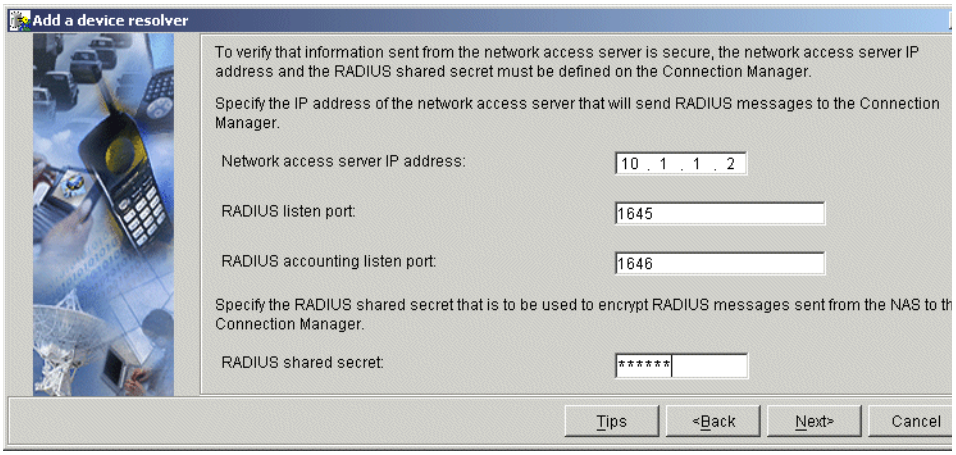
Connection Manager: cn=m23x3078.itso.ral.ibm.com,ou=linux,o=ibm,c=us

Device resolver name: NAS

Description: Win2000 Dial-Up Server

Figure 24-40 Device resolver name

Enter the IP address of the NAS server. These settings tell Connection Manager to trust the incoming RADIUS message from the NAS. Enter the RADIUS listner port; this should not be same that FreeRADIUS is listening to (1812) if they are in the same machine. Enter the port number for the RADIUS accounting messages; this has to be same that you defined in NAS. Also enter a secret which you entered in NAS and click **Next**.


Add a device resolver

To verify that information sent from the network access server is secure, the network access server IP address and the RADIUS shared secret must be defined on the Connection Manager.

Specify the IP address of the network access server that will send RADIUS messages to the Connection Manager.

Network access server IP address: 10 . 1 . 1 . 2

RADIUS listen port: 1645

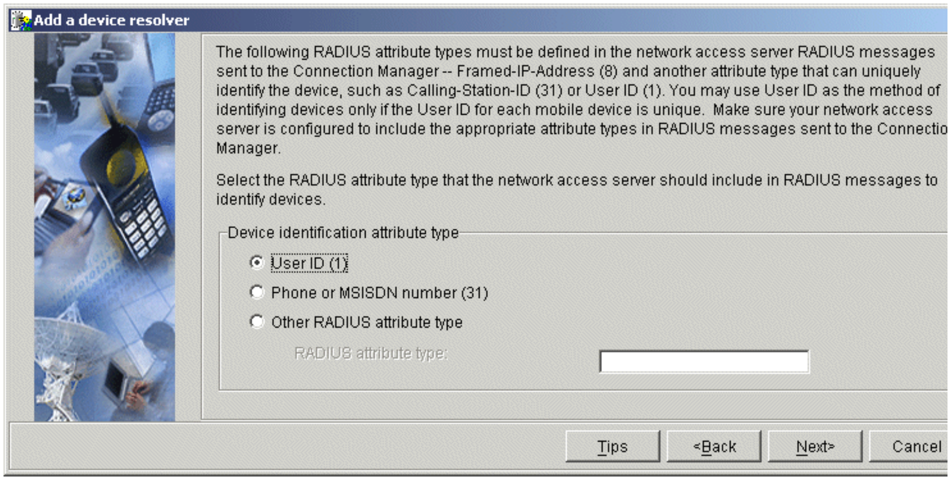
RADIUS accounting listen port: 1646

Specify the RADIUS shared secret that is to be used to encrypt RADIUS messages sent from the NAS to th Connection Manager.

RADIUS shared secret: ******

Figure 24-41 RADIUS port information

Select **User ID(1)** as the device identification attribute type and click **Next** (Figure 24-42).



The following RADIUS attribute types must be defined in the network access server RADIUS messages sent to the Connection Manager -- Framed-IP-Address (8) and another attribute type that can uniquely identify the device, such as Calling-Station-ID (31) or User ID (1). You may use User ID as the method of identifying devices only if the User ID for each mobile device is unique. Make sure your network access server is configured to include the appropriate attribute types in RADIUS messages sent to the Connectio Manager.

Select the RADIUS attribute type that the network access server should include in RADIUS messages to identify devices.

Device identification attribute type

● User ID (1)
○ Phone or MSISDN number (31)
○ Other RADIUS attribute type

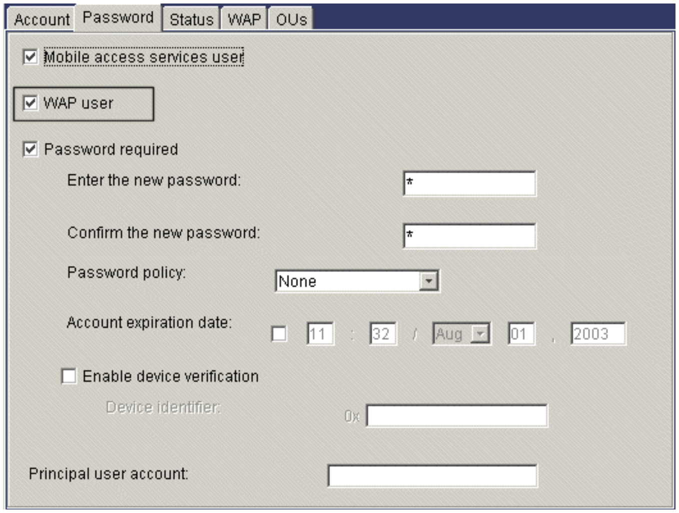
RADIUS attribute type:

Figure 24-42 Device identification attribute

You can configure Connection Manager also to forward the RADIUS messages; it is not needed to enable it in this scenario. Click **Finish**.

24.25.3 Creating testuser account

Add a new user using Gatekeeper. Enter `testuser` as the username, and `password` as the password. When the user is created, double-click the **User** icon, select **testuser,** and click the **Properties** button. Check the **WAP user** setting and click **OK** (Figure 24-43).



☑ Mobile access services user

☑ WAP user

☑ Password required

Enter the new password: *

Confirm the new password: *

Password policy: None

Account expiration date: □ 11 : 32 / Aug 01 , 2003

□ Enable device verification

Device identifier: 0x

Principal user account:

Figure 24-43 User settings

24.26 Testing the configuration

Dialing to NAS

To test the configuration, you can use a laptop with the null-modem cable connected directly to the NAS.

Select your null-modem connection from the network and dial-up connections, right-click and select **Connect**.

Windows asks for the username and password; enter `testuser` as the username and `password` as the password, and click the **Connect** button.

Your computer should now “dial” to the NAS. You will see status messages during dial if the last sentence is similar to `Registering your computer to network connection was successful.`

If there are any error dialogs, you should check your configurations, especially RADIUS settings such as the IP addresses and secrets.

If you started FreeRADIUS in debug mode, you can see possible error messages in the console. Otherwise, see the log files for any errors.

WAP browsing

When the dial-up connection has successfully established, start your WAP emulator, and check that the emulator is configured to use Connection Manager as the WAP gateway. Enter a URL for any WAP enabled Internet or intranet site. Your WAP emulator should now open the page without prompting you for a password.

If you receive errors such as `HTTP Error 403 Forbidden`, Connection Manager has not recognized you. You should first check that the user in Connection Manager has rights to use WAP. Then you should check the RADIUS accounting settings in Connection Manager, and the NAS such as IP addresses and secret.

You can see Connection Manager logs for any RADIUS error messages. Also see the NAS servers System event log for any remote access errors.

C

TAI plug-in example

This appendix describes how to write a simple Trust Association Interceptor (TAI) plug-in to utilize:

- Connection Manager HTTP headers
- Connection Manager session database
- Single sign-on

24.27 Overview

The TAI plug-in provided by Connection Manager uses sender IP address as a device identification. When reverse proxy is used to handle HTTP requests, the sender IP address is always the address of the proxy, not the original user. In this case, TAI cannot validate the user and SSO is failed.

This TAI plug-in extracts device identification from header that Connection Manager adds to every request. The header usually contains the IP address of the original user, which is then used to look up the username. This way you can enable SSO when using reverse proxy and WAP proxy.

Now, lets see what happens when the HTTP request arrives to the WebSphere Application server (WAS) where our TAI plug-in is installed:

1. HTTP request arrives from Connection Manager to the WebSphere Application Server.
2. WebSphere Application Server checks if there are any LTPA token cookies in the request. If not, WebSphere Application Server calls the `isTargetInterceptor()` method of the TAI plug-in.
3. TAI checks that the required headers exists in request.
4. WebSphere Application Server calls the `getAuthenticatedUsername()` method of the TAI plug-in.
5. TAI extracts the device identification from the header, usually from X-IBM-PVC-Client-id.
6. TAI queries the Connection Manager session database to get the user ID of the device.
7. TAI gives the user ID to the WebSphere Application Server.
8. WebSphere Application Server creates a LTPA token for the user and continues processing the request.

> **Note:** This example is meant for testing purposes only; you should not use this example in a public production environment.

24.28 Installation and configuration

First, you need to compile the class in Example 24-11, using for example WebSphere Application Developer (WSAD).

Note: You need some WebSphere libraries during compilation; if you are not using WSAD, make sure that those libraries are included in the classpath.

When the class is compiled, export the class to a jar file, for example, `itsotai.jar`. Then place the jar file to a place that is included in the WebSphere classpath, for example, to `/WebSphere/AppServer/lib`.

Example 24-11 TaiExample.java

```
package com.ibm.itso.wecm.example;

import java.sql.*;
import javax.sql.*;
import java.util.*;

import javax.naming.NamingException;
import javax.servlet.http.HttpServletRequest;

import com.ibm.websphere.security.*;

public class TaiExample extends WebSphereBaseTrustAssociationInterceptor
implements TrustAssociationInterceptor
{

   private String header = null;
   private String datasource = null;
   private String via = null;

   public String getAuthenticatedUsername(HttpServletRequest req) throws
WebTrustAssociationUserException
   {

      Connection connection = null;
      String userName = null;

      try
      {
         String deviceKey = req.getHeader(header);
         if (deviceKey == null || deviceKey.equals(""))
            throw new WebTrustAssociationUserException("Requested header:
"+header+" not found!");

         javax.naming.Context namingContext = new
javax.naming.InitialContext();
         DataSource ds = (DataSource) namingContext.lookup(datasource);

         connection = ds.getConnection();
         PreparedStatement statement = connection.prepareStatement("SELECT * 
FROM WG.ACTIVESESSIONATTRIBUTE WHERE DEVKEY=?");
         statement.setString(1, deviceKey);

         ResultSet rs = statement.executeQuery();

         if (rs.next())
         {
            userName = rs.getString("USERID");
         };
         rs.close();

         if (userName == null || userName.equals(""))
            throw new WebTrustAssociationUserException("User not found from 
session database for device(" + deviceKey + ")");
      }
      catch (NamingException n)
      {
         throw new WebTrustAssociationUserException(n.toString());
      }
      catch (SQLException e)
      {
         throw new WebTrustAssociationUserException(e.toString());
      }
      finally
      {
         if (connection != null)
            try
            {
               connection.close();
            }
            catch (Exception e)
            {
            }
      }

      return userName;

   }

   public boolean isTargetInterceptor(HttpServletRequest req) throws 
WebTrustAssociationException
   {
      return checkHeaders(req);
   }

   public void validateEstablishedTrust(HttpServletRequest req) throws 
WebTrustAssociationFailedException
```

```
   {
      if(!checkHeaders(req))
         throw new WebTrustAssociationFailedException("Requested headers not 
found");
   }

   public void cleanup()
   {
   }

   public int init(String props)
   {
      PropertyResourceBundle bundle = null;

      try
      {
         bundle = (PropertyResourceBundle) ResourceBundle.getBundle(props);
      }
      catch (Exception e)
      {
         e.printStackTrace();
         return -1;
      }

      this.header = bundle.getString("com.itso.wecm.example.tai.header");
      this.datasource = 
bundle.getString("com.itso.wecm.example.tai.datasource");
      this.via = bundle.getString("com.itso.wecm.example.tai.via");


      setVersion("ITSO WECM TAI example 1.0");

      return 0;
   }

   private boolean checkHeaders(HttpServletRequest req) throws 
WebTrustAssociationFailedException
   {
      try
      {
         if (req.getHeader(header) == null || req.getHeader(header).equals(""))
            return false;

         // If via is null or empty, we do not check the header
         if(via==null || via.equals(""))
            return true;

         String HTTP_Via = req.getHeader("Via");
         if (HTTP_Via == null || HTTP_Via.equals(""))
            return false;

         HTTP_Via = HTTP_Via.substring(HTTP_Via.indexOf(' ') + 1);
         HTTP_Via = HTTP_Via.substring(0, HTTP_Via.indexOf(' '));

         if (via.equals(HTTP_Via))
            return false;

         return true;
      }
      catch (Exception e)
      {
         throw new WebTrustAssociationFailedException(e.toString());
      }

   }

}
```

24.28.1 Configuring trustedservers.properties

Open the `trustedservers.properties` file for editing. The file is located in `/WebSphere/AppServer/properties`.

In Example 24-12 is a typical trustedservers.properties file, which is pre-configured to use a WebSeal TAI. If the file is not alike in your environment, it is okay, all you need to do is make sure that trust association is enabled, and to add lines that are in boldface.

Example 24-12 trustedservers.properties

```
# Trust Association Properties
# IBM WebSphere Application Server, Version 4.0, 2001
com.ibm.websphere.security.trustAssociation.enabled=true

#Use this property to specify the types of reverse proxy
#servers that will be loaded at runtime
com.ibm.websphere.security.trustassociation.types=webseal,itsotai

#For each type of reverse proxy servers specified in
#com.ibm.websphere.security.trustassociation.types,
#specify the class file that implements the associated
#interceptor for it.
com.ibm.websphere.security.trustassociation.webseal.interceptor=com.ibm.ws.secu
rity.web.WebSealTrustAssociationInterceptor
com.ibm.websphere.security.trustassociation.itsotai.interceptor=com.ibm.itso.we
cm.example.TaiExample
```

```
#Optionally, specify a properties file for any of the
#reverse proxy servers type specified above. The properties file
#must end with ".properties". e.g. webseal36.properties. However,
#do not include this extension as shown below. Moreover, you can only
#do this if the interceptor class extends
WebSphereBaseTrustAssociationInterceptor
#and both init() and cleanup() methods were  implemented. The init()
#method should read and parse the contents of the properties file.
com.ibm.websphere.security.trustassociation.webseal.config=webseal
com.ibm.websphere.security.trustassociation.itsotai.config=itsotai
```

24.28.2 Configuring itsotai.properties

Next, you need to create a property file for our TAI plug-in. Create a new file to /WebSphere/AppServer/properties and name it to itsotai.properties (Example 24-13).

Example 24-13 itsotai.properties

```
#
# This is a property file of a sample WECM TAI plug-in
#

com.itso.wecm.example.tai.header=X-IBM-PVC-Client-id

# Name of the datasource which provides access to the WECM session database
com.itso.wecm.example.tai.datasource=jdbc/WECM

# Hostname of the WECM proxy. If empty, via header is not checked
com.itso.wecm.example.tai.via=m23x3078
```

The first setting is for the HTTP header to look for a device identification. Next, is a JNDI name of the datasource used to connect to the Connection Manager session database.

The last one is the IP address or the hostname of the Connection Manager gateway as printed in through the HTTP header. In this example only the WAP proxy through the header will be parsed; however, you can leave this entry blank to indicate that you want to disable the through header checking.

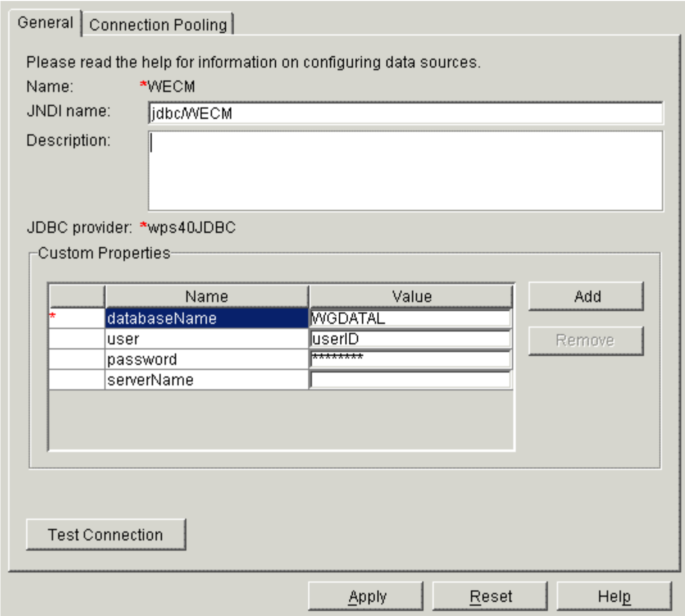
24.28.3 Configuring a Connection Manager datasource

In order to connect to the Connection Manager database, a datasource needs to be created. Open the WebSphere Administration client and expand the **Resources -> JDBC Providers** folder. If there is no JDBC provider that you can use, you need to create one. When creating a new JDBC provider, you need to know the name of the JDBC driver implementation class.

When the appropriate the JDBC provider is found and created; expand it and right-click **Data Sources** and select **New.**

Fill-in the required information (Figure 24-44):

- Datasource name
- Datasource JNDI name (must be same than in itsotai.properties)
- Database name
- User ID
- Password
- Server name



Please read the help for information on configuring data sources.

Name: *WECM

JNDI name: jdbc/WECM

Description:

JDBC provider: *wps40JDBC

Custom Properties

	Name	Value
*	databaseName	WGDATAL
	user	userID
	password	********
	serverName	

Figure 24-44 Datasource setup

Note: If you are using DB2, make sure that the JDBC service is started on DB2 server.

D

WebSphere Everyplace Suite samples

The follow section provides samples that may be used when integrating Connection Manager V5 into a WebSphere Everyplace Suite environment.

The details provided are:

- Sample LDIF file containing LDAP changes required by WebSphere Everyplace Suite when using Connection Manager
- Sample wgmgrd.conf

Sample LDIF

The following LDIF definition may be used to add the required configuration to a WebSphere Everyplace Suite directory server prior to installing Connection Manager V5. If viewing the softcopy of this redbook, simply cut and paste the sample text into a text file on the LDAP host. After making the necessary changes to the relevant variable, import the completed file into your LDAP.

```
##########################################################
#                                                        #
# WES LDIF Definition for WECM 5.0                       #
#                                                        #
#  If installing WECM in the WES r2.1.5 domain, create a #
#    file similar to this, then use the 'ldapmodify'     #
#    command to add the entries to the WES LDAP.         #
#                                                        #
#  Modify the file below:                                #
#    $hostname$ - Change all occurrences of $hostname$ to#
#      the short (unqualified) hostname of the machine   #
#      where WECM is installed.                          #
#    $suffix$ - Change all occurrences of $suffix$ to the#
#      TCP/IP domain of the machine where WECM is        #
#      installed. Use the '.' to separate each part of   #
#      domain, and prefix it with 'dc='. See example     #
#      below.                                            #
#    $fullhostname$ = Change all occurrences of          #
#      $fullhostname$ to the fully qualified hostname of #
#      the machine where WECM is installed.              #
#                                                        #
#  For example, if WECM is installed on wecm.yourco.com, #
#    then:                                               #
#      $hostname$ = wecm                                 #
#      $suffix$ = dc=yourco,dc=com                       #
#      $fullhostname$ = wecm.yourco.com                  #
#                                                        #
#  After creating the LDIF file, use the 'ldapmodify'    #
#    command to to add the entries to LDAP. While not a  #
#    requirement, it is suggested that you run the       #
#    ldapmodify command on the LDAP machine.             #
#                                                        #
#    ldapmodify -h <ldaphost> -a -D cn=<ldapid>          #
#      -w <ldapid pwd> -f <this file>                    #
#                                                        #
#  After adding the entries to LDAP, use the program     #
#    'dmt' to locate 'sys=SDP->sys=ewg'. Open 'sys=ewg'  #
#    and add the following to 'ibm-hostedonsystemref':   #
#      dc=$hostname,$suffix$                             #
#    Using the above example, 'ibm-hostedonsystemref'    #
#      would be set to:                                  #
#        dc=wecm,dc=yourco,dc=com                      #
#    If you install WECM on more than one machine, add   #
#      the other machines to 'ibm-hostedonsystemref'     #
#                                                      #
########################################################

########################################################
#                                                      #
# If you have already installed WES components on the    #
# WECM machine, then leave the following lines as        #
# comments. Otherwise, uncomment them, and modify them   #
# per the above instructions.                            #
#                                                      #
########################################################

# dn: dc=$hostname$,$suffix$
# dc: $hostname$
# host: $fullhostname$
# objectclass: eComputerSystem
# objectclass: dcObject
# objectclass: cimLogicalElement
# objectclass: cimManagedElement
# objectclass: cimManagedSystemElement
# objectclass: eSystem
# objectclass: top

dn: serviceName=svcewg1,dc=$hostname$,$suffix$
version: 5.0
servicename: svcewg1
objectclass: ibm-SdpComponent
objectclass: ibm-ServiceComponentPtr
objectclass: eService
objectclass: cimLogicalElement
objectclass: cimManagedElement
objectclass: cimManagedSystemElement
objectclass: top
ibm-sdpcomponenttype: ewg
ibm-serviceusesserviceref: serviceName=svcewg_svr1,dc=$hostname$,$suffix$
ibm-serviceusesserviceref: serviceName=svcewg_kpr1,dc=$hostname$,$suffix$
ibm-serviceusesserviceref: serviceName=svcewg_ard1,dc=$hostname$,$suffix$
ibm-serviceusesserviceref: serviceName=svcewg_tac1,dc=$hostname$,$suffix$
ibm-serviceusesserviceref: serviceName=svcewg_rad1,dc=$hostname$,$suffix$
ibm-serviceusesserviceref: serviceName=svcewg_dal1,dc=$hostname$,$suffix$
ibm-serviceusesserviceref: serviceName=svcewg_lan1,dc=$hostname$,$suffix$
ibm-serviceusesserviceref: serviceName=svcewg_mob1,dc=$hostname$,$suffix$
ibm-serviceusesserviceref: serviceName=svcewg_scr1,dc=$hostname$,$suffix$
ibm-serviceusesserviceref: serviceName=svcewg_rnc1,dc=$hostname$,$suffix$
ibm-serviceusesserviceref: serviceName=svcewg_sms1,dc=$hostname$,$suffix$
ibm-serviceusesserviceref: serviceName=svcewg_smt1,dc=$hostname$,$suffix$
```

```
ibm-serviceusesserviceref: serviceName=svcewg_snp1,dc=$hostname$,$suffix$

dn: serviceName=svcewg_svr1,dc=$hostname$,$suffix$
version: 5.0
ibm-serviceusedbyserviceref: serviceName=svcewg1,dc=$hostname$,$suffix$
servicename: svcewg_svr1
objectclass: ibm-SdpComponent
objectclass: ibm-ServiceComponentPtr
objectclass: eService
objectclass: cimLogicalElement
objectclass: cimManagedElement
objectclass: cimManagedSystemElement
objectclass: top
ibm-sdpcomponenttype: ewg_svr

dn: serviceName=svcewg_kpr1,dc=$hostname$,$suffix$
version: 5.0
ibm-serviceusedbyserviceref: serviceName=svcewg1,dc=$hostname$,$suffix$
servicename: svcewg_kpr1
objectclass: eService
objectclass: ibm-SdpComponent
objectclass: ibm-ServiceComponentPtr
objectclass: cimLogicalElement
objectclass: cimManagedElement
objectclass: cimManagedSystemElement
objectclass: top
ibm-sdpcomponenttype: ewg_kpr

dn: serviceName=svcewg_ard1,dc=$hostname$,$suffix$
version: 5.0
ibm-serviceusedbyserviceref: serviceName=svcewg1,dc=$hostname$,$suffix$
servicename: svcewg_ard1
objectclass: ibm-ServiceComponentPtr
objectclass: ibm-SdpComponent
objectclass: eService
objectclass: cimLogicalElement
objectclass: cimManagedElement
objectclass: cimManagedSystemElement
objectclass: top
ibm-sdpcomponenttype: ewg_ard

dn: serviceName=svcewg_tac1,dc=$hostname$,$suffix$
version: 5.0
ibm-serviceusedbyserviceref: serviceName=svcewg1,dc=$hostname$,$suffix$
servicename: svcewg_tac1
objectclass: ibm-SdpComponent
objectclass: eService
objectclass: ibm-ServiceComponentPtr
objectclass: cimLogicalElement
```

```
objectclass: cimManagedElement
objectclass: cimManagedSystemElement
objectclass: top
ibm-sdpcomponenttype: ewg_tac

dn: serviceName=svcewg_rad1,dc=$hostname$,$suffix$
version: 5.0
ibm-serviceusedbyserviceref: serviceName=svcewg1,dc=$hostname$,$suffix$
servicename: svcewg_rad1
objectclass: ibm-ServiceComponentPtr
objectclass: ibm-SdpComponent
objectclass: eService
objectclass: cimLogicalElement
objectclass: cimManagedElement
objectclass: cimManagedSystemElement
objectclass: top
ibm-sdpcomponenttype: ewg_rad

dn: serviceName=svcewg_dal1,dc=$hostname$,$suffix$
version: 5.0
ibm-serviceusedbyserviceref: serviceName=svcewg1,dc=$hostname$,$suffix$
servicename: svcewg_dal1
objectclass: eService
objectclass: ibm-ServiceComponentPtr
objectclass: ibm-SdpComponent
objectclass: cimLogicalElement
objectclass: cimManagedElement
objectclass: cimManagedSystemElement
objectclass: top
ibm-sdpcomponenttype: ewg_dal

dn: serviceName=svcewg_lan1,dc=$hostname$,$suffix$
version: 5.0
ibm-serviceusedbyserviceref: serviceName=svcewg1,dc=$hostname$,$suffix$
servicename: svcewg_lan1
objectclass: ibm-ServiceComponentPtr
objectclass: ibm-SdpComponent
objectclass: eService
objectclass: cimLogicalElement
objectclass: cimManagedElement
objectclass: cimManagedSystemElement
objectclass: top
ibm-sdpcomponenttype: ewg_lan

dn: serviceName=svcewg_mob1,dc=$hostname$,$suffix$
version: 5.0
ibm-serviceusedbyserviceref: serviceName=svcewg1,dc=$hostname$,$suffix$
servicename: svcewg_mob1
objectclass: ibm-SdpComponent
```

```
objectclass: ibm-ServiceComponentPtr
objectclass: eService
objectclass: cimLogicalElement
objectclass: cimManagedElement
objectclass: cimManagedSystemElement
objectclass: top
ibm-sdpcomponenttype: ewg_mob

dn: serviceName=svcewg_scr1,dc=$hostname$,$suffix$
version: 5.0
ibm-serviceusedbyserviceref: serviceName=svcewg1,dc=$hostname$,$suffix$
servicename: svcewg_scr1
objectclass: eService
objectclass: ibm-SdpComponent
objectclass: ibm-ServiceComponentPtr
objectclass: cimLogicalElement
objectclass: cimManagedElement
objectclass: cimManagedSystemElement
objectclass: top
ibm-sdpcomponenttype: ewg_scr

dn: serviceName=svcewg_rnc1,dc=$hostname$,$suffix$
version: 5.0
ibm-serviceusedbyserviceref: serviceName=svcewg1,dc=$hostname$,$suffix$
servicename: svcewg_rnc1
objectclass: ibm-ServiceComponentPtr
objectclass: eService
objectclass: ibm-SdpComponent
objectclass: cimLogicalElement
objectclass: cimManagedElement
objectclass: cimManagedSystemElement
objectclass: top
ibm-sdpcomponenttype: ewg_rnc

dn: serviceName=svcewg_sms1,dc=$hostname$,$suffix$
version: 5.0
ibm-serviceusedbyserviceref: serviceName=svcewg1,dc=$hostname$,$suffix$
servicename: svcewg_sms1
objectclass: ibm-SdpComponent
objectclass: ibm-ServiceComponentPtr
objectclass: eService
objectclass: cimLogicalElement
objectclass: cimManagedElement
objectclass: cimManagedSystemElement
objectclass: top
ibm-sdpcomponenttype: ewg_sms

dn: serviceName=svcewg_smt1,dc=$hostname$,$suffix$
version: 5.0
```

```
ibm-serviceusedbyserviceref: serviceName=svcewg1,dc=$hostname$,$suffix$
servicename: svcewg_smt1
objectclass: ibm-SdpComponent
objectclass: eService
objectclass: ibm-ServiceComponentPtr
objectclass: cimLogicalElement
objectclass: cimManagedElement
objectclass: cimManagedSystemElement
objectclass: top
ibm-sdpcomponenttype: ewg_smt

dn: serviceName=svcewg_snp1,dc=$hostname$,$suffix$
version: 5.0
ibm-serviceusedbyserviceref: serviceName=svcewg1,dc=$hostname$,$suffix$
servicename: svcewg_snp1
objectclass: eService
objectclass: ibm-ServiceComponentPtr
objectclass: ibm-SdpComponent
objectclass: cimLogicalElement
objectclass: cimManagedElement
objectclass: cimManagedSystemElement
objectclass: top
ibm-sdpcomponenttype: ewg_snp
```

wgmgrd.conf

The following sample may be used to create the wgmgrd.conf file that is required prior to installing WECM V5 in a WebSphere Everyplace Suite environment. Be sure to replace the relevant variables with your own settings:

```
#
# IBM Wireless Gateway Admin Configuration File
#
# CAUTION: DO NOT EDIT THIS FILE DIRECTLY.
#          All changes should be made through Gatekeeper
#
objectclass             = wlCfg
objecttype              = wlCfg
cn                      = Access Manager
dsstype                 = 1
dssloc                  = ldaphost
basedn                  = $suffix$
version                 = 1.0
ldapadmin               = cn=<ldapid>
ldappasswd              = <ldapid pwd>
ldapport                = 389
ldapporttwo             = 389
```

```
translate               = 0
ldaptimeout             = 30
ldapsecmode             = 1
maxresourcehits         = 100
maxsearchhits           = 500
wesldaparch             = 1
ldapcryptpass           = 1
srchusermaps            = 0
wloumaps                = organizationalUnit
wlusermaps              = ePerson
wgmgrsdKdbFilename      = /usr/lpp/wireless/wgmgrsd.trusted.kdb
wgmgrsdSthFilename      = /usr/lpp/wireless/wgmgrsd.trusted.sth
wgmgrsdV2timeout        = 100
wgmgrsdV3timeout        = 600
wgmgrsdTrafficTrace     = 0
remoteroot              = 1
onlysecureconns         = 0
wgmgrd_log              = 0
wgmgrsd_log             = 0
usermanagement          = 0
acctsystype             = 1
wpsstoretype            = 1
dbmstype                = 0
dbmstype2               = 0
wapmakeauthhdr          = 0
```

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 592. Note that some of the documents referenced here may be available in softcopy only.

- *The eNetwork Wireless Solution,* SG24-5299
- *Understanding LDAP*, SG24-4986
- *A Comprehensive Guide to Virtual Private Networks, Volume III: Cross-Platform Key and Policy Management*, SG24-5309
- *IBM WebSphere Everyplace Server Service Provider and Enable Offerings: Enterprise Wireless Applications*, SG24-6519
- IBM WebSphere Everyplace Access V4.3, SG24-7015-01

Online resources

These Web sites and URLs are also relevant as further information sources:

- WebSphere Everyplace Connection Manager site

  `http://www.ibm.com/software/pervasive/products/mobile_sols/connection_manager.shtml`
- White papers, tools and resources

  `http://www.ibm.com/software/pervasive/products/library/wireless_gateway.shtml`
- Everyplace Wireless Gateway

  `http://www.ibm.com/software/pervasive/products/mobile_sols/wireless_gateway.shtml`
- Pervasive Computing Software

  `http://www.ibm.com/software/pervasive/tech/downloads/`
- Authentication and encryption

  `http://www.rsasecurity.com/rsalabs/faq/`
- WAP Forum

  `http://www.wapforum.org/what/technical.htm`

- Java

  http://www.javasoft.com
- DB2 Technical Library

  http://www.ibm.com/software/data/db2/library/
- Internet Engineering Task Force RFC pages

  http://www.ietf.org/rfc.html
- WAP Forum conformance release

  http://www.openmobilealliance.org/wapdownload.html

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

IBM WebSphere Everyplace Connection Manager V5 Handbook

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

IBM WebSphere Everyplace Connection Manager V5 Handbook

Provide secure wireless access to your e-business applications

Reduce data transmission costs over wireless networks

Deploy mobile access and messaging services

This IBM Redbook helps you plan and implement wireless solutions to access backend resources such as databases, application servers, and other legacy applications from wireless devices. This redbook deals with IBM WebSphere Everyplace Connection Manager provided functions to enable businesses to make a smooth transition to the wireless Web.

The information provided in this redbook targets Business-to-Employee (B2E) enterprise applications, but most of the scenarios presented apply to Business-to-Consumer (B2C) applications as well. In this redbook, you will find examples and scenarios showing ways to extend your enterprise applications to a broad range of mobile devices such as WAP phones, PDAs, and laptops using wireless and dial-up connections. Enterprise applications that can be accessed include WebSphere Application Server applications, portal applications, WebSphere Everyplace Access, MQ Everyplace, Relational Database Synchronization, and others.

In this redbook you will also find sample scenarios showing ways to install and administrate Connection Manager using the Gatekeeper tool to configure support for WAP devices, HTTP services, Mobility Clients, messaging, clustering, roaming, and the available security features such as authentication, data encryption, and digital certificates.

A basic knowledge of wireless and Web technologies is assumed.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7049-00 ISBN 0738498297