

Unified Messaging for WebSphere Voice Response



Web Services Voicemail API

Version 4.2

Note

Before using this information and the product it supports, read the general information under “Notices” on page 43.

First edition (May 2008)

This edition applies to Version 4 Release 2 of IBM® Unified Messaging for WebSphere® Voice Response (program number 5724-I07) and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

© Copyright International Business Machines Corporation 1994, 2008. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	v	Set Email Notification	19
Tables	vii	Get Email Notification	20
About this book	ix	Get Folder Counts	21
Who should use this book	ix	Set Mailbox Presence Status	22
How to use this book	x	Get Mailbox Presence Status	24
Typographic conventions	x	Delete Message	26
Accessibility	x	Forward Message	26
Notes on terminology	xi	Save Message	29
Where to find more information	xi	Get Message Information	30
Useful Web sites	xi	Change Password	32
Making comments on this book	xii	Validate Password	32
Chapter 1. Introduction	1	Set Preferences	33
What is a Web service?	1	Get Preferences	34
SOAP	1	Set Subscriber Information	35
Web Services Description Language	3	Get Subscriber Information	37
Chapter 2. Using the Unified Messaging		Chapter 4. Mailbox audio content	39
Web services API	9	Get Message	39
Creating a Web service client proxy from a		Create New Voice Message	40
WSDL file	9	Notices	43
Unified Messaging Web services architecture	10	Trademarks	44
Installing Unified Messaging Web services	12	Glossary	47
Deploying Unified Messaging Web services		Product documentation	61
on WebSphere Application Server	12	WebSphere Voice Response base software	61
Deploying Unified Messaging Web services		WebSphere Voice Response related products	62
on Apache Tomcat	14	WebSphere Voice Server; Use with IBM	
Example	15	WebSphere Voice Response for AIX	62
Changing logging settings dynamically	15	AIX	63
Chapter 3. Mailbox access	17	Computers running AIX	63
Set Call Forward Number	18	Brooktrout Fax	63
Get Call Forward Number	18	Index	65

Figures

1.	Example HTTP message header	2	4.	Example SOAP response message	3
2.	Example SOAP message	2	5.	Unified Messaging Web services	
3.	Example HTTP response message header	2	architecture		11

Tables

1.	WSDL file structure	4	21.	29
2.	setCallForwardNumber()	18	22.	getMessageInformation()	30
3.	setCallForwardNumber() response	18	23.	getMessageInformation() response	30
4.	getCallForwardNumber()	18	24.	changePassword()	32
5.	getCallForwardNumber() response	19	25.	changePassword() response	32
6.	setEmailNofication()	19	26.	validatePassword()	33
7.	setEmailNofication() response	20	27.	validatePassword() response.	33
8.	getEmailNotification()	21	28.	setPreferences()	33
9.	getEmailNotification() response.	21	29.	setPreferences() response	34
10.	getFolderCounts()	21	30.	getPreferences()	34
11.	getFolderCounts() response	22	31.	getPreferences() response	35
12.	setMailboxPresenceStatus()	23	32.	setSubscriberInformation()	35
13.	setMailboxPresenceStatus() response	24	33.	setSubscriberInformation() response	37
14.	getMailboxPresenceStatus()	24	34.	getSubscriberInformation()	37
15.	getMailboxPresenceStatus() response	25	35.	getSubscriberInformation() response	37
16.	deleteMessage()	26	36.	getMessage()	39
17.	deleteMessage() response.	26	37.	getMessage() response.	40
18.	forwardMessage()	26	38.	createVoiceMessage()	40
19.	forwardMessage() response	28	39.	createVoiceMessage() response	42
20.	saveMessage()	29			

About this book

This book tells you how to make features of IBM Unified Messaging for WebSphere Voice Response available in a Web Services environment.

IBM Unified Messaging for WebSphere Voice Response is an application that runs on IBM WebSphere Voice Response for AIX® Version 4.2

Throughout this book, we refer to IBM Unified Messaging for WebSphere Voice Response as **Unified Messaging**. We refer to IBM WebSphere Voice Response for AIX as **WebSphere Voice Response**, except where we need to distinguish between particular releases of the product.

The purpose of this book is to document the Unified Messaging functionality currently available using a generic voice mail API for use in a Web Services environment. The functionality is split into two main sections, mailbox access and mailbox audio.

The functionality described in Chapter 3, “Mailbox access,” on page 17:

- Provides essential functionality required to demonstrate read access to a voice mail mailbox.
- Allows for the interrogation and updating of certain aspects of mailbox information and user status.

The functionality described in Chapter 4, “Mailbox audio content,” on page 39 enriches the API to provide a more complete suite of interactions with the back end.

Who should use this book

The primary audience for this document includes, but is not limited to:

- Web Services developers
- Web Services testers
- Messaging service provider support personnel

To use this book successfully, you should already be familiar with **Unified Messaging, WebSphere Voice Response**. Although not essential, some knowledge of Web Services and how to develop them would also be useful.

How to use this book

This book is comprised of an introduction to Web Services and how to develop them, followed by reference topics including detailed information describing the **Unified Messaging** Web Services API.

Typographic conventions

This book uses the following typographic conventions:

boldface

Identifies an **item** in a WebSphere Voice Response window. The item could be a keyword, an action, a field label, or a pushbutton. Whenever one of the steps in a procedure includes a word in boldface, look for an item in the window that is labeled with that word.

boldface italics

Are used for emphasis. *Take extra care* wherever you see bold italics!

italics Identify one of the following:

- *New terms* that describe Unified Messaging components or concepts. A term printed in italics is usually followed by its definition.
- *Parameters* for which you supply the actual names or values.
- References to other *books*.

monospace

Text that you type in an AIX window. Because AIX is case sensitive, make sure you type the uppercase and lowercase characters exactly as shown.

Accessibility

The Unified Messaging application is developed to run on WebSphere Voice Response and provide telephone access to business data and services. In this way, Unified Messaging provides accessibility for people who cannot access the data and services by using regular Web pages or traditional graphical interfaces. These telephone user interfaces are fully accessible to people who are blind or have low vision and, if speech recognition is implemented, to people with mobility impairments or limited hand use. Speech recognition capability can be provided by products such as IBM WebSphere Voice Server. In addition, support for users of Telephony Devices for the Deaf (TDD) is provided as part of the WebSphere Voice Response product.

With Unified Messaging and WebSphere Voice Response you can perform many application development and system administration tasks using a text editor or line commands – these are accessible if you use a screen reader

product to interface with them. Also, the default settings of the WebSphere Voice Response graphical user interface can be changed to produce large fonts and high contrast colors. Details of how to use these accessibility features can be found in the *WebSphere Voice Response for AIX: User Interface Guide*. Alternatively, application development can be done using Java™ or VoiceXML development tools supplied by IBM and third parties.

You can also use a screen-reader product to access the Unified Messaging and WebSphere Voice Response publications in HTML format (for details of availability refer to “Product documentation” on page 61 at the back of this book).

Notes on terminology

- A glossary of commonly-used terms is provided at the end of this book
- The full product name of *WebSphere Voice Response for AIX with DirectTalk® Technology* is generally abbreviated in this book to *WebSphere Voice Response*.
- The term *IBM System p5™* is generically used in this book to refer both to PCI-based RS/6000® computers and to appropriate models of the IBM System p5™ and pSeries® ranges. (Consult your IBM representative for details of models that are supported for use with WebSphere Voice Response.)

Where to find more information

Apart from this and other books in the Unified Messaging library, your main source of information is likely to be the WebSphere Voice Response for AIX library. Becoming familiar with the WebSphere Voice Response for AIX library will help you accomplish tasks more quickly.

Details of all available books in the Unified Messaging library, as well as other product related documentation can be found in “Product documentation” on page 61.

Useful Web sites

The following Web sites are useful sources of information about WebSphere Voice Response and related products:

IBM WebSphere voice products

Select the **Products** link on the Pervasive Computing Software home page at <http://www.ibm.com/software/pervasive>

VoiceXML Version 2.1 specification

<http://www.w3.org/TR/voicexml21>

VoiceXML Version 2.0 specification

<http://www.w3.org/TR/2001/WD-voicexml20-20011023>

where to find more information

CCXML Version 1.0 specification

<http://www.w3.org/TR/ccxml>

Making comments on this book

If you especially like or dislike anything about this book, feel free to send us your comments.

You can comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book. Please limit your comments to the information in this book only and the way in which the information is presented. Speak to your IBM representative if you have suggestions about the product itself.

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

You can get your comments to us quickly by sending an e-mail to **idrcf@hursley.ibm.com**. Alternatively, you can mail your comments to:

User Technologies
IBM United Kingdom Laboratories,
Mail Point 095, Hursley Park,
Winchester, Hampshire, SO21 2JN, United Kingdom

Please ensure that you include the book title, order number, and edition date.

Chapter 1. Introduction

This chapter introduces Web services and highlights their importance in the provision of Web-based distributed applications necessary for business integration and inter-operability.

What is a Web service?

A Web service is a software application function packaged as a reusable component that can be used in a business process and accessed over a network, typically the Internet or a company intranet. Web services Application Programming Interfaces (APIs) allow loose coupling between the interacting software agents. This is achieved by ensuring that services have no dependencies of the state of other services and have a well-defined interface that is not tied to any particular implementation.

Service requests and responses need only include information related to the business function—information on the implementation details of the APIs, file formats, and the operating systems used are not be required, but each message or response must contain sufficient information to allow the message or response to be processed.

To ensure the flexibility and genuine inter-operability required for open business integration, all participating software systems must use agreed standard communications protocols and interaction mechanisms.

Typically:

- Messages used by Web services adhere to the XML-based SOAP protocol.
- Web services are identified by a unique Uniform Resource Identifier (URI).
- Web service interfaces are described in the XML-based Web Services Description Language (WSDL).

For more information on Web services, refer to:

<http://www.ibm.com/developerworks/webservices>

SOAP

Originally, SOAP was an acronym for Simple Object Access Protocol, but this is now generally regarded as misleading because SOAP is not an access protocol, is far from simple, and does not transmit objects or object data. SOAP is a lightweight protocol for carrying XML messages across a network. A soap message is actually an XML document. It is independent of the:

- Programming language used to create it

- Network
- Transport medium

The underlying transport protocol used with SOAP (the binding) is usually HTTP, which is synchronous, but other transport protocols such as JMS can also be used when asynchronous messaging is required. No binding information is included in the message. SOAP version 1.1, which is the most common implementation used for Web services, supports only HTTP POST requests.

```
POST /UMAPI/services/VMailAPISOAP HTTP/1.1
Host: localhost:9080
Accept: application/soap+xml,multipart/related,text/*
User-Agent: IBM WebServices/1.0
Cache-Control: no-cache
Pragma: no-cache
SOAPAction: "http://www.ibm.com/UM/VMailAPI/validatePassword"
Connection: Keep-Alive
Content-Type: text/xml; charset=utf-8
Content-Length: 438
Date: Wed, 19 Mar 2008 11:05:50 GMT
```

Figure 1. Example HTTP message header

```
<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:q0="http://www.ibm.com/UM/VMailAPI/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header/>
  <soapenv:Body>
  <q0:getSubscriberInformation>
    <userID>987654</userID>
    <password>1234</password>
  </q0:getSubscriberInformation>
  <soapenv:Body>
  <soapenv:Envelope>
```

Figure 2. Example SOAP message

```
HTTP/1.1 200 OK
```

Figure 3. Example HTTP response message header

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header/>
  <soapenv:Body>
    <getSubscriberInformationResponse xmlns="http://www.ibm.com/UM/VMailAPI/">
    <getSubscriberInformationResponse xmlns="">
      <name>John Doe</name>
      <language>en_US</language>
      <emailAddress>john.doe@us.ibm.com</emailAddress>
      <passwordExpiryDate>20100101</passwordExpiryDate>
      <result>successful</result>
    </getSubscriberInformationResponse>
  </getSubscriberInformationResponse>
</soapenv:Body>
</soapenv:Envelope>

```

Figure 4. Example SOAP response message

SOAP message types

There are two ways in which SOAP messages are used:

Document exchange

Based on an arbitrary formats agreed by the sender and receiver. The structure and syntax of the messages is defined by an XML schema. Messages are typically exchanged in a predefined sequence rather like a conversation.

Remote procedure call (RPC)

The message identifies in a single child element a particular operation or response and includes in child elements the parameter names or values of the arguments for the function or method being passed. By convention the response operation is identified by the name used in the request but with the string "Response" appended, for example requestNameResponse, where requestName was the request identifier.

Related information

SOAP specifications

<http://www.w3.org/TR/soap/>

SOAP resources

http://www.xml.org/xml/resources_focus_soap.shtml

Web Services Description Language

Web Services Description Language (WSDL) is an XML-based specification for describing web services and their implementation. A WSDL definition allows the abstract model of a service to be mapped on to a particular message format and network protocol.

WSDL allows a service provider to publish in a service broker UDDI service registry descriptions that include sufficient information for a service requester to discover and invoke the service dynamically in an automatic process.

WSDL allows service interfaces to be defined in a standard way that is independent of the implementation platform and programming language used. A service interface in WSDL can be converted into any programming language. Tools such as IBM Rational Application Developer can create Java stubs to access Web services and generate Java Web service clients from WSDL files. Axis WSDL2Ws tool can create stubs and generate Web service clients in C and C++, and Microsoft's Wsd.exe tool can generate code in C#, Visual Basic, Visual J#, and JScript. Tools that generate code in other languages such as Perl from a WSDL file are also available

A WSLD file has the following general structure:

Table 1. WSDL file structure

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
Namespace definitions
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://www.ibm.com/UM/VMailAPI/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="VMailAPI"
targetNamespace="http://www.ibm.com/UM/VMailAPI/"
  <wsdl:types>
    <xsd:schema targetNamespace="http://www.ibm.com/UM/VMailAPI/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

Data type definitions in the form of an XML schema

```
    <xsd:element name="validatePassword">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="userID" type="xsd:string"/>
          <xsd:element name="password" type="xsd:string"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  :
</xsd:schema>
</wsdl:types>
```


Table 1. WSDL file structure (continued)

Message definitions

```
<wsdl:message name="validatePasswordRequest">
  <wsdl:part element="tns:validatePassword" name="parameters"/>
</wsdl:message>
:
<wsdl:portType name="VMailAPI">
```

Operations available to clients of the service, and references to the messages involved

```
<wsdl:operation name="validatePassword">
  <wsdl:input message="tns:validatePasswordRequest"/>
  <wsdl:output message="tns:validatePasswordResponse"/>
</wsdl:operation>
:
</wsdl:portType>
<wsdl:binding name="VMailAPISOAP" type="tns:VMailAPI">
```

Associations between an operation (or portTypegroup of operations) and a protocol binding

```
<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="validatePassword">
  <soap:operation soapAction="http://www.ibm.com/UM/VMailAPI/validatePassword"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
:
<wsdl:service name="VMailAPI">
```

Declaration of the service and the ports on which it is available

```
<wsdl:port binding="tns:VMailAPISOAP" name="VMailAPISOAP">
  Association of a particular network address and a binding
  <soap:address location="http://localhost:9080/UMAPI/services/VMailAPISOAP"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

Namespace definitions

The URL format used ensures that the name of each namespace referenced is unique and that standard definitions are used.

Data type definitions

Allows customized data types used in the interface to be defined in an XML schema.

Message definitions

An operation can involve zero through three exchanges of information: a request, a response, and a fault. A single exchange is defined in a message

Operations

The <portType> element contains the definitions for one or more operations, each of which references the messages involved in an <operation> element. Operations, which must have an endpoint, can be one of four types:

One-way

The endpoint receives a message.

Request-response

The endpoint receives a message and sends a correlated message.

Solicit-response

The endpoint sends a message and receives a correlated message. Currently, this communications pattern is not supported by a WSDL binding.

Notification

The endpoint sends a message. Currently, this communications pattern is not supported by a WSDL binding.

Bindings

Associate an operation (or <portType> group of operations) with a protocol binding. The <binding> <operation> elements define the way in which particular operations are 'marshaled' on to a specific binding. A binding can be specified as document or literal (for RPC operation) over HTTP. See "SOAP message types" on page 3.

Service definitions

Identify the actual location of the service that implements the interface described in the referenced binding. Each port is specific to a particular binding.

To facilitate reuse and enable a service requester to choose between different implementations of a service interface, a WSDL file can be split into multiple documents: a service interface document and one or more different service implementation documents.

Related information**WSDL 1.1 specification**

<http://www.w3.org/TR/wsd1/>

SOAP resources

http://www.xml.org/xml/resources_focus_soap.shtml

XML schema

<http://www.w3.org/XML/Schema>

Apache Axis WSDL tools

<http://ws.apache.org/axis/cpp/arch/WSDL2Ws.html>

Microsoft .NET framework WSDL tools

[http://msdn2.microsoft.com/en-us/library/7h3ystb6\(VS.71\).aspx](http://msdn2.microsoft.com/en-us/library/7h3ystb6(VS.71).aspx)

Chapter 2. Using the Unified Messaging Web services API

The WSDL file supplied with Unified Messaging PTF contains all the information necessary to expose your Unified Messaging voice messaging system as a Web service. See Chapter 3, “Mailbox access,” on page 17 and Chapter 4, “Mailbox audio content,” on page 39 for details of the messages and their parameters used in the Unified Messaging Web service API operations.

Using tools such as IBM Rational Application Developer you can create stubs to access Web services and generate Web service clients from the supplied WSDL file.

Creating a Web service client proxy from a WSDL file

To create a voice messaging Web service client proxy from the supplied Unified Messaging WSDL file using IBM Rational Application Developer:

1. In Rational Application Developer, create a new project of the type .
2. Import file VMai1API.wsdl from installation directory into the newly created project.
3. Select the file VMai1API.wsdl in the project.
4. Click the right-hand mouse button to display a pop-up menu.
5. Click **Web Services - > Generate Client** to generate the Web client files required to access Unified Messaging Web services running on a Web server.
6. In the Web Services window, select **Java proxy**. If you also want to test and monitor the service when it has been created, select the **Test the Web service** and **Monitor the Web service** check boxes. Click **Next**.
7. In the Web Service Selection window displayed, leave the default URI for the VMai1API.wsdl, and click **Next**.
8. In the Client Environment Configuration window displayed:
 - a. Select **Web** to generate code for Web client (.jsp files). To generate Java code (for a Java client, a Java Application client, or an EJB client), select the appropriate **Client type** from the list displayed.
 - b. Select an existing project of the appropriate type the **Client project** list. For example, if you are generating code for Web client, Select an existing Web project.
 - c. Leave the default **EAR project**, and click **Next**.

9. In the Web Service Proxy window, select the security configuration you require. Leave the other defaults as they are and click **Next** or **Finish** to generate the code.
10. Run the client application.

Unified Messaging Web services architecture

Figure 5 on page 11 illustrates the various APIs , including the Web services API.

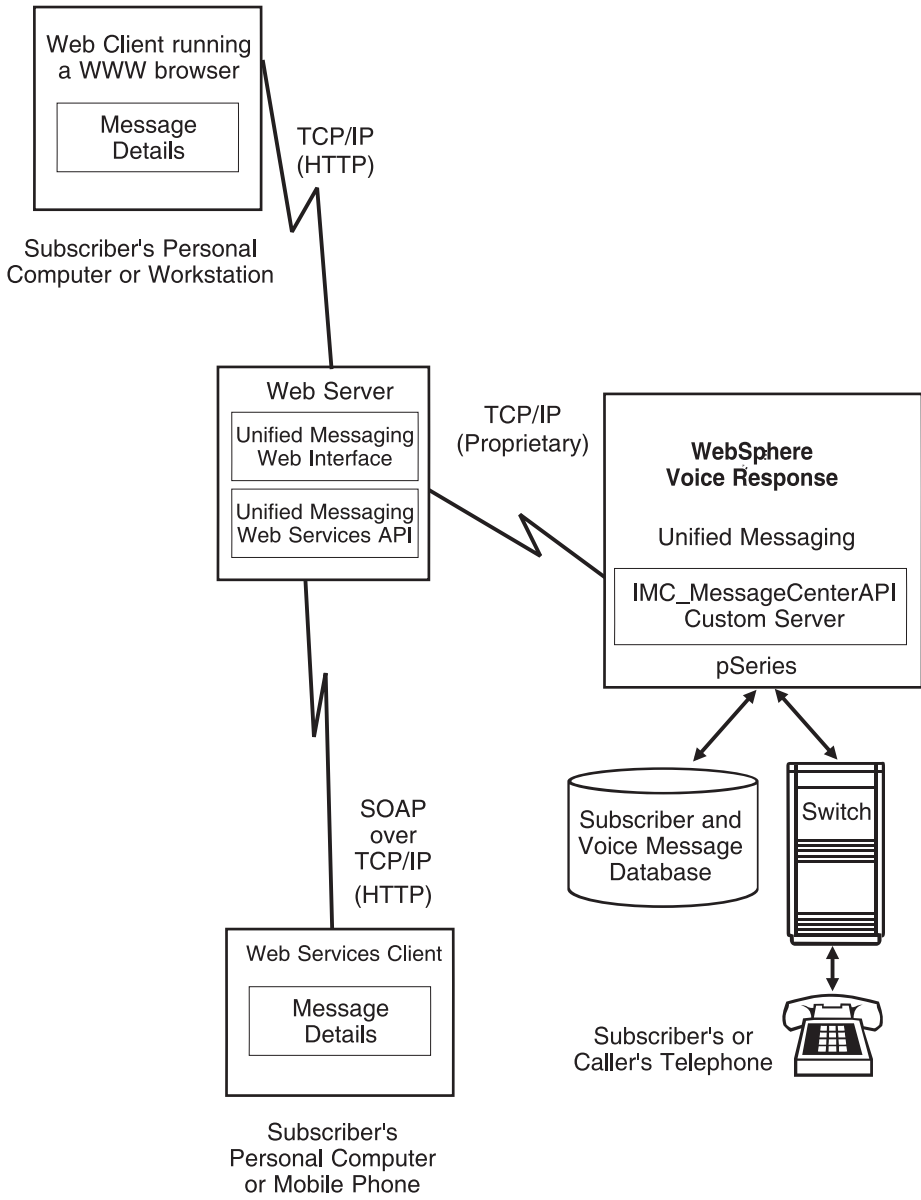


Figure 5. Unified Messaging Web services architecture

The new Web services API complements the other interfaces.

Installing Unified Messaging Web services

To install Unified Messaging Web services:

1. Install WebSphere Voice Response PTF U817184. Refer to “Installing PTFs for WebSphere Voice Response” in the *WebSphere Voice Response for AIX: Installation* for more information on how to do this.
2. Use `smitty` to install fileset `dirTalk.UM.web`, Fix Level: 4.2.0.136. Refer to “Starting installation of WebSphere Voice Response” in the *WebSphere Voice Response for AIX: Installation* for details of the procedure to follow.

Deploying Unified Messaging Web services on WebSphere Application Server

To deploy your Unified Messaging Web service on WebSphere Application Server:

1. Copy the file `$CUR_DIR/Web_Services/UMAPI.war` to any directory on the machine that has WebSphere Application Server installed on it.
2. Login to the WebSphere Application Server WebSphere Administrative Console.
 - a. From the Welcome window, click **Applications -> Install New Application**.
 - b. In the Preparing for the application installation window, type the path of the directory to where you copied the `UMAPI.war` file.
 - c. In the Context root field, type the name of the context root, `UMAPI` for example.
 - d. Click **Next**.
 - e. On the window displayed, click **Next**.
 - f. On the window displayed, click **Continue** to display the Select installation options window.
 - g. Click **Next** to accept the defaults and display the Map modules to servers window.
 - h. Click **Next** to accept the defaults and display the Map virtual hosts for Web modules window.
 - i. Click **Next** to accept the defaults and display the Summary window.
 - j. Click **Finish**.
 - k. Click **Save to Master Configuration**.
 - l. Click **Save** again
3. Set the JVM properties:
 - a. From the Welcome window, click **Application servers -> server_name -> Server Infrastructure: Java and Process Management -> Process Definition > Additional Properties: Java Virtual Machine ->**
 - b. In the General Properties window, type 1024 in the **Initial Heap Size** field and 1024 in the **Maximum Heap Size** field.

- c. Click **Apply**.
 - d. From the General Properties window, click **Additional Properties: Java Virtual Machine >Additional Properties: Custom Properties > New**
An empty Configuration Properties window is displayed.
 - e. Type WUMA_ADDRESS in the **Name** field.
 - f. Type the IP address of the machine on which you are running Unified Messaging in the **Value** field. It is set to null by default.
 - g. Optionally, type a description in the **Description** field.
 - h. Click **Apply**.
 - i. The Unified Messaging API default port is 25121. Optionally, to specify a different TCP/IP port for use by Unified Messaging, type WUMA_PORT in the **Name** field and the port in the **Value** field.
 - j. Type WUMA_LOG_LEVEL in the **Name** field.
 - k. Type a digit between 0 and 3 to specify the level of logging required:
 - 0 Only severe error messages are logged.
 - 1 Severe and warning errors messages are logged.
 - 2 Severe, warning and information messages are logged.
 - 3 Severe, warning, information and detailed messages are logged.

To specify the most detailed logging, type 3. This property is set to 0 by default.
 - l. Optionally, to specify a property file that is read every time a new request comes into the system, so it can be used to change logging levels without stopping and starting the application server, type WUMA_PROPERTY_FILE in the **Name** field and the full path of the property file in the **Value** field. This property is set to null by default.
If specified, the referenced property file can contain the following settings:


```
WUMA_LOG_FILE
WUMA_LOG_LEVEL
WUMA_ADDRESS
WUMA_PORT
```

These settings have the same meanings as the JVM property settings and will take precedence over any JVM property settings.
 - m. Click **OK** to save your changes and close the window.
4. Shutdown WebSphere Application Server by clicking on **Start -> Programs -> IBM WebSphere -> Application Server v6 -> Profiles -> default -> Stop the server**.
 5. Start WebSphere Application Server up again clicking on **Start -> Programs -> IBM WebSphere -> Application Server v6 -> Profiles -> default -> Start the server**.

6. To start the Unified Messaging Web services application:
 - a. Log in to the WebSphere Application Server WebSphere Administrative Console.
 - b. From the Welcome window, click **Applications > Enterprise Applications**.
 - c. Select the name of the context root that you specified in step 2c, for example **UMAPI**, and click **Start**.

Deploying Unified Messaging Web services on Apache Tomcat

To deploy your Unified Messaging Web service on an Apache Tomcat web server:

1. Shutdown Tomcat from the `%CATALINA_HOME%\bin\` directory by running the **shutdown.bat** command.
2. Copy the file `$CUR_DIR/Web_Services/UMAPI.war` to the **webapps** directory on the machine that has the Tomcat web server installed on it. (The default installation directory is
C:\Program Files\Apache Software Foundation\Tomcat 4.1\webapps)
3. Edit the `%CATALINA_HOME%\bin\catalina.bat` by changing the following fields to the following settings:

Parameter

Value

-DWUMA_LOG_LEVEL

The level of logging required:

- 0 Fatal messages only are logged.
- 1 Fatal and error messages are logged.
- 2 Fatal, error, and warning messages are logged.
- 3 Fatal, error, warning, and information messages are logged.
- 4 Fatal, error, warning, information, and debug messages are logged.

For example, use the setting `-DWUMA_LOG_LEVEL=3` to log fatal, error, warning, and information messages. The default setting is 1.

-DWUMA_LOG_FILE

The file to be used for Tomcat output logging, for example:

`-DWUMA_LOG_FILE=c:\\temp\\soalog.txt` to log to the file `c:\\temp\\soalog.txt`. (The double back slashes are required.)

-DWUMA_ADDRESS

The TC/PIP address of the Unified Messaging system that is running the Unified Messaging API, for example,
WUMA_ADDRESS=umapi.myco.com

It is set to null by default.

4. Ensure that the heap size of the machine running the Unified Messaging API is set to a suitable size by editing the Tomcat environment variable JAVA_OPTS setting in the catalina.bat or catalina.sh file. For example to increase the heap size to 1024 Megabytes, use the setting
JAVA_OPTS=-Xmx1024m.
5. Start Tomcat up again from the %CATALINA_HOME%\bin\ directory by running the **startup.bat** command.

Example

The following example is based on lines from the Catalina.bat file on a Windows system running Apache Tomcat version 4.1:

```
set JAVA_OPTS=-Xmx1024m
-DWUMA_LOG_LEVEL=3
-DWUMA_LOG_FILE=c:\\temp\\soalog.txt
-DWUMA_ADDRESS=machine0.myco.com
```

Changing logging settings dynamically

To allow logging settings to be changed on a call per call basis, use a properties file to configure the JVM. In cases of extremely heavy load, the file access to the properties file on each call may prove to be an issue, but under normal use this should not be the case. If a property file is not used, the JVM has to be stopped and started in order to change logging settings.

To use a property file on window set the following JVM property:

```
-DWUMA_PROPERTY_FILE=c:\\temp\\properties.txt
```

Include lines similar to the following in the referenced properties.txt file:

```
WUMA_LOG_LEVEL=1
WUMA_LOG_FILE=c:\\temp\\soa111log.txt
WUMA_ADDRESS=machine0.myco.com
```

Chapter 3. Mailbox access

The following set of APIs is intended for use by a client application to interrogate the contents of a voice messaging mailbox. The functionality is intended for typical day to day end-user functionality such as checking for and listening to voice messages.

The interface has also been designed generically such that the underlying voice messaging platform remains platform independent.

Logon

“Change Password” on page 32

“Validate Password” on page 32

Messages

“Get Folder Counts” on page 21

“Delete Message” on page 26

“Forward Message” on page 26

“Save Message” on page 29

“Get Message Information” on page 30

Mailbox Settings

“Set Call Forward Number” on page 18

“Get Call Forward Number” on page 18

“Set Email Notification” on page 19

“Get Email Notification” on page 20

“Set Mailbox Presence Status” on page 22

“Get Mailbox Presence Status” on page 24

Subscriber Preferences

“Set Preferences” on page 33

“Get Preferences” on page 34

“Set Subscriber Information” on page 35

“Get Subscriber Information” on page 37

Message audio functionality available to web services is described in Chapter 4, “Mailbox audio content,” on page 39.

Set Call Forward Number

The `setCallForwardNumber()` function is intended as a mechanism to set a mailbox's associated call forward number.

Table 2. setCallForwardNumber()

Parameter	Description	Type
userID	The reference to the user ID	String
password	The password associated with the user ID	String
phoneNumber	The new number to which the call forward number is to be set (maximum length 256)	String

Table 3. setCallForwardNumber() response

Response	Description	Type
Result	Successful	String
	Unrecognized userID	
	Invalid password	
	Password expired	
	Account locked Invalid	
	Timeout Unsupported	
	Error	
	Failed Local Policy	
	No FMFM number supplied	
	FMFM number too short	
FMFM number too long		

Get Call Forward Number

The `getCallForwardNumber()` function is intended to retrieve the current call forward number associated with a given mailbox.

Table 4. getCallForwardNumber()

Parameter	Description	Type
userID	The reference to the user ID	String
password	The password associated with the user ID	String

Table 5. *getCallForwardNumber()* response

Response	Description	Type
result	Successful Unrecognised userID Invalid password Password expired Account locked Invalid Timeout Unsupported System Error	String
phoneNumber	Call forward number	String

Set Email Notification

The `setEmailNotification()` function is intended as a mechanism to enable or disable notification to an email account of new voice messages. It also provides the ability to specify that the voice message be attached to the notification email, and if this is the case if the voice message should be kept in the mailbox or deleted.

Table 6. *setEmailNotification()*

Parameter	Description	Type
userID	The reference to the user ID	String
password	The password associated with the user ID	String
action	enable or disable - if enable, notifications are sent to emailAddress	String
emailAddress	The email address to which notifications are to be sent.	String
attachment	true or false - if true, message is sent as an attachment with notification.	String
deletion	true or false - if true, message is deleted from mailbox when sent as attachment.	String
format	The file audio format of the attachment, which can be wav or au.	String

Table 7. setEmailNofication() response

Response	Description	Type
result	Successful	String
	Invalid	
	Unrecognized user ID	
	Invalid password	
	Account locked	
	Timeout	
	Unsupported action - known but not supported	
	Invalid action - not known	
	Unsupported type - known but not supported	
	Invalid type - not known	
	Invalid deletion - not known	
	Invalid subtype - not valid for type	
	Invalid address - not valid for type	
	Invalid attachment - not valid for type	
	Invalid format	
	Unsupported	

If the action parameter is set to enable, an email address must be specified. This is used to set both the subscriber's email and his or her notification email address. If action is set to disable, the notification email address is cleared and the subscriber's email address is set to the value specified in emailAddress. If attachment is set to "" (an empty string), only action and emailAddress are processed. If attachment is set to true, deletion and format must be specified. If attachment is set to false, the deletion parameter (the value of which is assumed to be true) and theformat parameter are ignored.

Get Email Notification

The getEmailNotification() function is intended as a mechanism to determine if notification to an email account of new voice messages is enabled. It also provides the ability to determine if the voice message will be attached to the notification email and if this is the case if the voice message will be kept in

the mailbox or deleted.

Table 8. getEmailNotification()

Parameter	Description	Type
userID	The reference to the user ID	String
password	The password associated with the user ID	String

Table 9. getEmailNotification() response

Response	Description	Type
result	Successful Invalid Unrecognised user ID Invalid password Account locked Timeout Unsupported	String
action	enable or disable - if enable, notifications are sent to emailAddress.	String
emailAddress	The email address to which notifications are to be sent.	String
attachment	true or false - if true, message is sent as an attachment with notification.	String
deletion	true or false - if true, message is deleted from mailbox when sent as attachment.	String
format	The file audio format of the attachment, which can be wav or au.	String

Get Folder Counts

The `getFolderCounts()` function is used to return a summary of the number of messages left in a given user's mailbox or mailboxes.

Table 10. getFolderCounts()

Parameter	Description	Type
userID	The reference to the user ID	String
password	The password associated with the user ID	String

Table 11. *getFolderCounts()* response

Response	Description	Type
result	Successful Unrecognised userID Invalid password Password expired Account locked Invalid Timeout Unsupported System error	String
folder[x] folderName (String) folderCount (Integer)	The new message details	sequence

Note: This functionality does assume that there is only one mailbox associated with a given user.

-1 should denote unsupported by platform

Fixed list of folder names:

folderName	Meaning
saved	The number of saved messages
deleted	The number of messages marked as deleted
outgoing	The number of outgoing messages
fax	The number of outgoing messages

Set Mailbox Presence Status

The `setMailboxPresenceStatus()` function is intended as a mechanism to set a mailboxes availability settings / greeting to a given presence status.

The function is intended to change the current settings to an alternative, and is likely to be associated with an internet messaging style status change, for example when an IM status is changed, a client would be able to make a similar change to the voicemail status.

Table 12. *setMailboxPresenceStatus()*

Parameter	Description	Type
userID	The reference to the user ID	String
password	The password associated with the user ID	String
presenceStatus	<p>The new presence status, which can be one of the following values:</p> <p>inoffice_accept In the office and accepting messages.</p> <p>inoffice_notaccept In the office, but not accepting messages.</p> <p>outofoffice_accept Available, but working away from the office</p> <p>outofoffice_notaccept Out of the office and not accepting messages.</p> <p>unknown</p>	String
presenceText	Presence Text representation	String

Table 13. *setMailboxPresenceStatus()* response

Response	Description	Type
Result	Successful	String
	Unrecognised userID	
	Invalid password	
	Password expired	
	Invalid	
	Account locked	
	Timeout	
	Unsupported	
	System error	
	Invalid presence	
Unsupported presence		

Note: The inclusion of the presenceText field will provide an optional mechanism to pass additional presence information to a platform.

Get Mailbox Presence Status

The `getMailboxPresenceStatus()` function is intended as a mechanism to retrieve a mailbox's availability settings / greeting and associated it with a generic presence status.

Table 14. *getMailboxPresenceStatus()*

Parameter	Description	Type
UserID	The reference to the user ID	String
Password	The password associated with the user ID	String

Table 15. *getMailboxPresenceStatus()* response

Response	Description	Type
Result	Successful Unrecognised userID Invalid password Password expired Invalid Account locked Timeout Unsupported System error	String
presenceStatus	Presence Status. Valid values are: inoffice_accept In the office and accepting messages. inoffice_notaccept In the office, but not accepting messages. outofoffice_accept Available, but working away from the office outofoffice_notaccept Out of the office and not accepting messages. unknown	String
presenceText	Presence text representation	String

Note: The presenceText is included to allow additional presence specific text to be passed from the platform.

Delete Message

The deleteMessage() function is to delete a specified voice message from a given users mailbox.

Table 16. deleteMessage()

Parameter	Description	Type
userID	The reference to the user ID	String
password	The password associated with the user ID	String
messageID	The message ID to delete	String

Table 17. deleteMessage() response

Response	Description	Type
result	Successful Unrecognised userID Invalid userID Invalid password Password expired Account locked Invalid Timeout Unsupported System error Invalid Message ID Message not found Not possible on this message	String

Forward Message

The forwardMessage() function is to be used forward a specified voice message to another mailbox or phone number.

Table 18. forwardMessage()

Parameter	Description	Type
userID	The reference to the user ID	String

Table 18. `forwardMessage()` (continued)

Parameter	Description	Type
<code>password</code>	The password associated with the user ID	String
<code>messageID</code>	The message ID to forward	String
<code>destination</code>	Destination to which the message is to be forwarded	String
<code>acknowledgement</code>	Acknowledgement enabled: 1 Acknowledgement not required 2 Acknowledgement required	Integer
<code>urgency</code>	The message urgency: 1 normal 2 urgent 3 emergency	Integer
<code>privacy</code>	Modify current privacy status: 1 public 2 private	Integer
<code>commentData</code>	Data (0 if not applicable)	Hex/Bin
<code>commentFormat</code>	("" if not applicable) wav-pcm8 wav-pcm16 wav-mulaw wav-alaw au-mulaw au-alaw msgsm	String
<code>commentLength</code>	Comment length (0 if no comment data provided)	Integer

Table 19. *forwardMessage()* response

Response	Description	Type
result	Successful Unrecognized userID Invalid password Password expired Account locked Invalid Timeout Unsupported Invalid message ID Invalid destination Invalid urgency Invalid privacy Invalid acknowledge System error Cannot forward private message Message could not be forwarded Message not found Invalid CommentData Invalid CommentFormat Unsupported CommentFormat	String

If `commentLength` is set to 0 or less, `commentData` and `commentFormat` are ignored.

At present only wav (8 bit .wav) comments are supported.

Save Message

The `saveMessage()` function is to be used to save a specified voice message from a users mailbox for use at a later time.

Table 20. saveMessage()

Parameter	Description	Type
userID	The reference to the user ID	String
password	The password associated with the user ID	String
messageID	The message ID to save	String
folder	The folder to save message into. Valid values are: new – currently not supported saved deleted – currently not supported outgoing – currently not supported	String

Table 21.

Response	Description	Type
result	Successful Unrecognized userID Invalid userID Invalid password Password expired Account locked Timeout Unsupported Invalid Message ID Invalid folder Unsupported folder Message not found	String
expiryDate	The date when the saved message will expire (if known). Empty string if not known	String

Get Message Information

The `getMessageInformation()` function is used to return more detailed information of the actual voice messages contained within a given users mailbox.

Table 22. `getMessageInformation()`

Parameter	Description	Type
<code>userID</code>	The reference to the user ID	String
<code>password</code>	The password associated with the user ID	String
<code>folder</code>	See “Get Folder Counts” on page 21 for valid names, plus <code>all</code> to get details of messages in all supported folders.	String

Table 23. `getMessageInformation()` response

Response	Description	Type
<code>result</code>	Successful Unrecognised userID Invalid password Password expired Account locked Invalid Invalid folder Unsupported folder Timeout Unsupported No data found	String

Table 23. *getMessageInformation()* response (continued)

Response	Description	Type
message[x]		array
messageID	Message ID	String
format	mime type if known, otherwise an empty string	String
folder	Name – see “Get Folder Counts” on page 21.	String
timeMessageSent	The time that the message was sent: YYYYMMDDhhmmss / UTC	String
sender	The message sender YYYYMMDDhhmmss / UTC	String
timeMessageReceived	The time that the message was received: YYYYMMDDhhmmss / UTC	String
recipient	The message recipient	String
size	The message duration (in milliseconds for voice, bytes for FAX or EMAIL) Value of -1 if unknown or invalid.	Integer
urgency	The message urgency: 1 normal 2 urgent 3 emergency -1 unknown	Integer
privacy	The message is marked as private: 1 public 2 private -1 unknown	Integer
acknowledgement	Acknowledgement status: 1 acknowledge not required 2 acknowledge required 3 acknowledged 4 rejected -1 unknown	Integer

Change Password

The `changePassword()` function is intended as a mechanism to change the current password associated with a given `userID`.

Table 24. changePassword()

Parameter	Description	Type
<code>userID</code>	The reference to the user ID	String
<code>password</code>	The password associated with the user ID	String
<code>newPassword</code>	The replacement password for the user ID	String

Table 25. changePassword() response

Response	Description	Type
<code>result</code>	Successful Unrecognized user ID Invalid password Invalid Account locked Timeout Unsupported Invalid new password System error New password does not meet password rules Same as existing password Same as an old password	String

Validate Password

The `validatePassword()` function is intended as a mechanism to validate the users password.

This function is included for completeness as the `userID` and `password` are actually presented for every Unified Messaging web services API request due to the session-less nature of web service interfaces. It is anticipated that this function be used as a logon style request during client start up. Subsequent function calls are not dependent on the success of this request instance as no

state is recorded.

Table 26. *validatePassword()*

Parameter	Description	Type
userID	The reference to the mailbox login ID	String
password	The password associated with the mailbox LoginID	String

Table 27. *validatePassword()* response

Response	Description	Type
result	Successful Invalid Unrecognized user ID Invalid userID Invalid password Password expired Password locked Timeout Unsupported System error	String
passwordExpiryDate	The date when the password is due to expire	String (UTC)
passwordFailCount	The number of failed requests. -1 if not available.	Integer

Note: The passwordExpiryDate response is included to allow client applications the option to prompt if a password is due to expiry shortly.

Set Preferences

The setPreferences() function is used to set the value of one or more specified preference values.

Table 28. *setPreferences()*

Parameter	Description	Type
userID	The reference to the user ID	String
password	The password associated with the user ID	String

Table 28. *setPreferences()* (continued)

Parameter	Description	Type
preferences[]	Preference details. Set key to the value required. Currently, only the key <code>first_time_user</code> is supported along with value 0 or 1.	Array
key		String
value		String

Table 29. *setPreferences()* response

Response	Description	Type
result	Successful	String
	Invalid	
	Unrecognised user ID	
	Invalid userID	
	Invalid password	
	Password expired	
	Password locked	
	Timeout	
	Unsupported	
	System error	

Get Preferences

The `getPreferences()` function is used to return the value of one or more specified preference values.

Table 30. *getPreferences()*

Parameter	Description	Type
userID	The reference to the user ID	String
password	The password associated with the user ID	String
preferences[]	Preference details - set key to the value required. Currently, only the key <code>first_time_user</code> is supported. Value should be set to "".	Array
key		String
value		String

Table 31. *getPreferences()* response

Response	Description	Type
result	Successful Invalid Unrecognized user ID Invalid userID Invalid password Password expired password locked Timeout Unsupported System error	String
preferences[]	Preference details Key value specified Associated value.	Array
key	Currently, only the key <code>first_time_user</code> is supported.	String
value	If a key value is not found in the preferences, its value is returned as <code>**UNKNOWN**</code> .	String

Set Subscriber Information

The `setSubscriberInformation()` function is intended as a mechanism to set known information associated with a given user ID.

Table 32. *setSubscriberInformation()*

Parameter	Description	Type
userID	The reference to the user ID	String
password	The password associated with the user ID. Set to Null to ignore.	String
name	The mailbox owner's name.	String

Table 32. *setSubscriberInformation()* (continued)

Parameter	Description	Type	
language	The ISO language type:	String	
	For		Set to
	Belgian Dutch		nl_BE
	Belgian French		fr_BE
	Canadian French		fr_CA
	Danish		da_EU
	Finnish		fi_EU
	French		fr_EU
	German		de_EU
	Greek		el_EU
	Icelandic		is_EU
	Italian		it_EU
	Netherlands Dutch		nl_NL
	Norwegian		no_EU
	Portuguese		pt_EU
	Spanish		es_EU
	Swedish		sv_EU
	Swiss French		fr_CH
	Swiss German		de_CH
	Turkish		tr_TR
UK English	en_GB		
US English	en_US		
	Set to a numeric value if no mapping is possible.		
	Set to Null to ignore.		
emailAddress	Mailbox owner's email address. Set to Null to ignore The format is not checked.	String	

Table 33. *setSubscriberInformation()* response

Response	Description	Type
result	Successful Unrecognized user ID Invalid password Invalid Account locked Timeout Unsupported System error Invalid language	String

Get Subscriber Information

The `getSubscriberInformation()` function is intended as a mechanism to retrieve known information associated with a given user ID.

Table 34. *getSubscriberInformation()*

Parameter	Description	Type
userID	The reference to the user ID	String
password	The password associated with the user ID	String

Table 35. *getSubscriberInformation()* response

Response	Description	Type
result	Successful Unrecognized user ID Invalid password Invalid Account locked Timeout Unsupported System error	String

Table 35. *getSubscriberInformation()* response (continued)

Response	Description	Type	
name	The mailbox owner's name	String	
language	The ISO language type:	String	
	For		Set to
	Belgian Dutch		nl_BE
	Belgian French		fr_BE
	Canadian French		fr_CA
	Danish		da_EU
	Finnish		fi_EU
	French		fr_EU
	German		de_EU
	Greek		el_EU
	Icelandic		is_EU
	Italian		it_EU
	Netherlands Dutch		nl_NL
	Norwegian		no_EU
	Portuguese		pt_EU
	Spanish		es_EU
	Swedish		sv_EU
	Swiss French		fr_CH
	Swiss German		de_CH
	Turkish		tr_TR
UK English	en_GB		
US English	en_US		
	Set to a numeric value if no mapping is possible.		
emailAddress	Mailbox owner's email address.	String	
passwordExpiryDate	The date when the password will expire.	date	

Chapter 4. Mailbox audio content

This chapter describes the Unified Messaging functionality available to handle message audio in web services.

Get Message

The `getMessage()` function is used to retrieve information and audio associated with a given user mailbox and message ID. It is assumed that the `messageID` will have been derived from a prior `getMessageInformation()` request.

Table 36. getMessage()

Parameter	Description	Type
<code>userID</code>	The reference to the user ID	String
<code>password</code>	The password associated with the user ID	String
<code>messageID</code>	The message reference	Integer
<code>format</code>	Format required: wav-pcm8 wav-pcm16 wav-mulaw wav-alaw au-mulaw au-alaw msgsm	String

Table 37. *getMessage()* response

Response	Description	Type
result	Successful Unrecognised userID Invalid userID Invalid password Password expired Account locked Timeout Unsupported Invalid System error Invalid message ID Invalid format Unsupported format Message not found	Integer
length	The length of the data in bytes	Integer
data	Binary Data	Bin/Hex

Create New Voice Message

The `createVoiceMessage()` should be used to create and send a new voice message

It is envisaged that this mechanism could be used to create and deliver messages from external sources or even IVRs.

To be completed

Table 38. *createVoiceMessage()*

Parameter	Description	Type
userid	The reference to the user ID	String
password	The password associated with the user ID	String
destination	Destination to which the message is to be forwarded	String

Table 38. *createVoiceMessage()* (continued)

Parameter	Description	Type
acknowledgement	Acknowledgement enabled: <ol style="list-style-type: none"> 1 acknowledge not required 2 acknowledge required 	Integer
urgency	The message urgency: <ol style="list-style-type: none"> 1 Normal 2 Urgent 3 Emergency 	Integer
privacy	Modify current privacy status: <ol style="list-style-type: none"> 1 public 2 private 	Integer
format	The format of the message: wav-pcm8 wav-pcm16 wav-mulaw wav-alaw au-mulaw au-alaw msgsm	String
length	Length of data in bytes	Integer
data	Binary Data	Bin/Hex

Table 39. createVoiceMessage() response

Response	Description	Type
Result	Successful Unrecognised userID Invalid password Password expired Account locked Invalid Invalid destination Invalid format Timeout Unsupported	String
messageID	ID of created message	String

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:

The IBM Director of Licensing, IBM Corporation, North Castle Drive,
Armonk, NY 10504-1785, U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing, 2-31 Roppongi 3-chome
Minato-ku, Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make

improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact: IBM UK Limited, Department 88013, 4NW, 76/78 Upper Ground, London, SE1 9PZ, England. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

For country-specific notes on the use of WebSphere Voice Response, refer to the README file located in the directory /usr/lpp/dirTalk/homologation. The file name is in the format README_homologation.xxxx, where xxxx is the country/region identifier.

Trademarks

The following terms are trademarks of IBM Corporation in the United States or other countries or both:

AIX	BladeCenter	DirectTalk
IBM	pSeries	RS/6000
System p5	WebSphere	

Microsoft, Windows, Windows NT, 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 of Sun Microsystems, Inc. in the United States, other countries, or both.

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

Glossary

The following terms and abbreviations are defined as they are used in the context of Unified Messaging. You'll find a more complete glossary of WebSphere Voice Response terms in any of the WebSphere Voice Response books. If you still do not find the term or abbreviation you are looking for, see *IBM Dictionary of Computing, McGraw-Hill, 1994* or the *AIX: Topic Index and Glossary, SC23–2513*.

Numerics

6309 Digital Trunk Quad Adapter. See Digital Trunk Quad Adapter.

6310 Digital Trunk Extended Adapter. See Digital Trunk Extended Adapter.

A

action. See state table action.

addressee. In Unified Messaging, the subscriber to whom a message will be sent.

administrator profile. Data that describes a WebSphere Voice Response user. Information in an administrator profile includes ID, password, language preference, and access privileges.

alarm. Any condition that WebSphere Voice Response considers worthy of documenting with an error message. Strictly speaking, the term *alarm* should include only red (immediate attention) and yellow (problem situation) conditions, but it is also used to refer to green (a red or yellow message has been cleared) and white (informational) conditions.

Alias. An alternative number for a mailbox. Aliases allow several users access to a single mailbox using different numbers.

AMIS. See Audio Messaging Interchange Specification (AMIS).

AMIS-A. See AMIS analog.

AMIS analog (AMIS-A). The audio messaging interchange standard that specifies the use of DTMF tones to send control information, and analog signals for the message itself. It is the AMIS standard to which Unified Messaging conforms.

analog. Data in the form of continuously variable signals, such as voice or light signals.

announcement-only greeting. In voice mail, a greeting that does not give the caller an opportunity to leave a voice message.

application. See voice application.

application connectivity link (ACL). A service that transmits out-of-band information between WebSphere® Voice Response and the Siemens Hicom 300 switch.

application profile. Data that describes initial actions to be performed when the telephone is answered. Information in an application profile indicates to the channel process what state table to load.

ARTIC960RxD Quad Digital Trunk PCI Adapter. See Digital Trunk Quad Adapter

Audio Messaging Interchange Specification (AMIS). A set of voice messaging standards designed to enable messages from different voice messaging systems to be interchanged. See also AMIS analog.

audio name. The audible name that corresponds to a specific application profile ID and mailbox.

glossary

auto-attendant. Automated attendant. A voice application that answers incoming calls and asks the caller which number or other service they'd like. In Unified Messaging, an auto-attendant can be used to access subscribers' mailboxes when direct inward dialing (DID) is unavailable.

B

base WebSphere Voice Response system. The WebSphere Voice Response system that provides voice processing support for Unified Messaging.

C

call. Telephone call. Often used to mean a single runtime instance of a voice application.

call center. A central point of front-line telephone contact between an enterprise and its customers. Contrast with contact center.

call forwarding. The process of sending incoming calls to a telephone number other than the called number.

called number. The number that a caller dialed. This typically identifies the mailbox that is to receive a message in a Unified Messaging system.

called party. Any person, device, or system that receives a telephone call. Contrast with caller.

caller. (1) Any person, device, or system that makes a telephone call. (2) Often used to refer to any user of a voice application, even when WebSphere Voice Response has made an outbound call and the user is really the called number. (3) In Unified Messaging, any person who makes a telephone call to a subscriber.

CallPath. Software that provides basic computer-telephony integration (CTI) enablement and comprehensive CTI functionality. This includes access to, and management of, inbound and outbound telecommunications.

call transfer. A series of actions that directs a call to another telephone number.

CAS. See channel associated signalling (CAS).

central office. A telephone switching system that resides in the telephone service provider's network. There are different types of central office switches, depending upon the role of the switch within the telephone network. Commonly, a central office switch connects customer lines to other customer lines or trunks and is the point at which local subscriber lines terminate for switching to other lines or trunks.

CGI. See Common Gateway Interface (CGI).

channel. One of the 24 channels carried on a T1 trunk, or one of the 30 channels on an E1 trunk.

channel associated signalling (CAS). A method of communicating telephony supervisory or line signalling (on-hook and off-hook) and address signalling on T1 and E1 digital links. The signalling information for each traffic (voice) channel is transmitted in a signalling channel permanently associated with the traffic channel.

channel bank. A device that converts an analog line signal to a digital trunk signal.

channel number. The identifying number assigned to a licensed channel on the T1 or E1 trunk that connects WebSphere Voice Response to the switch, channel bank, or channel service unit.

channel process (CHP). The AIX® process that executes the logic of the state table; each active caller session has one active channel process.

clear message. A message displayed by WebSphere Voice Response to tell the operator that a red or yellow error message has been cleared.

Common Gateway Interface (CGI). An interface to programs that provide services on the Web.

computer-telephony integration (CTI). Connecting a computer to a telephone so that they share information and commands. Events from the computer can trigger events on the telephone system, and vice versa. The CTI connection can be on the desk top for one person

or on the switch for use by many people. CTI can include simple facilities such as call transfer and screen pops, as well as more complex services, such as intelligent call routing, load balancing, and coordinating multiple call centers.

concatenative text-to-speech. Text-to-speech that joins together samples of voice recordings to produce a more natural voice sound. See also formant text-to-speech.

contact center. A central point of front-line contact between an enterprise and its customers involving more than just the telephone. Contact centers can feature automated call handling, handle both inbound and outbound customer interactions, and conduct transactions using the Web or e-mail. Contrast with call center.

CTI. See computer-telephony integration (CTI).

custom server. A C language or C++ language program that provides data manipulation and local or remote data stream, database, or other services beyond those provided by the state table interface. Custom servers provide an interface between WebSphere Voice Response and business applications, functions, or other processes to give callers access to business information and voice processing functions such as speech recognition.

D

daemon. In the AIX operating system, a program that runs unattended to perform a standard service.

dB. Decibel.

DBIM. WebSphere Voice Response 's internal database manager.

DBS. WebSphere Voice Response 's database server.

DDI. See direct inward dialing (DID).

development system. A WebSphere Voice Response system that is not used to respond to or make *live* calls; it is used only to develop and test applications.

dial. To initiate a telephone call. In telecommunication, this action is performed to establish a connection between a terminal and a telecommunication device over a switched line.

dial by name. To press the keys that correspond to a subscriber's name rather than their telephone number or extension.

dial tone. An audible signal (call progress tone) that indicates that a device such as a PABX or central office switch is ready to accept address information (DTMF or dial pulses).

dialed number identification service (DNIS). A number supplied by the public telephone network to identify the number actually called. For example, two toll-free numbers might both be translated to the same real number. The DNIS information distinguishes which of the two numbers was dialed. DNIS can be used by CallPath Enterprise Client or WebSphere Voice Response to automatically select between several business database applications. Often used as a synonym for called number.

DID. See direct inward dialing (DID).

digital signal processing (DSP). A set of algorithms and procedures used to process electronic signals after their conversion to digital format. Due to the specific mathematical models required to perform this processing, specialized processors are generally used.

Digital Trunk Ethernet Adapter. The IBM® Digital Trunk Ethernet PCI Adapter. In WebSphere Voice Response this adapter is known as a DTEA. It allows you to connect an IBM System p5 computer that has a PCI bus directly to an IP network.

Digital Trunk Extended Adapter. The IBM ARTIC960RxD Quad Digital Trunk PCI Adapter. In WebSphere Voice Response this adapter is known as a DTXA. It allows you to connect directly to the telephony network an IBM System p5 computer that has a PCI bus; it doesn't need an external pack. Contrast with Digital Trunk Quad Adapter.

glossary

Digital Trunk Quad Adapter. An adapter that allows a Single Digital Trunk Processor or Multiple Digital Trunk Processor to be attached to a pSeries® computer that has a PCI bus, including the model 43P and model E20. Contrast with Digital Trunk Extended Adapter.

direct dial in (DDI). See direct inward dialing (DID).

direct inward dialing (DID). A service that allows outside parties to call directly to an extension of a switch. Known in Europe as direct dial in (DDI).

DirectTalk. An earlier name for WebSphere Voice Response, a voice processing system, bringing together telephone and data communications networks to use information stored in databases directly from a telephone.

DirectTalkMail. An earlier name for Message Center, which is now called Unified Messaging.

DirectTalkMail Digital (DTM-D). A digital protocol for exchanging voice messages very rapidly between Unified Messaging systems.

disconnect. To hang up or terminate a call.

distribution list. In voice mail, a list of subscribers to whom the same message can be sent.

DNIS. See dialed number identification service (DNIS).

double-trunking. See trombone.

DSP. See digital signal processing.

DTMF. See dual-tone multifrequency (DTMF).

DTEA. See Digital Trunk Ethernet Adapter.

DTQA. See Digital Trunk Quad Adapter.

dtuser. The name of the AIX account set up during the installation process for the use of all users of WebSphere Voice Response.

DTXA. See Digital Trunk Extended Adapter.

dual-line call transfer. A call transfer method in which the primary and secondary lines remain bridged until a call is completed. (Also known as **tromboning**; see trombone).

dual-tone multifrequency (DTMF). The signal sent by pressing one of the telephone keys. Each signal is composed of two different tones.

dynamic caller menu. A menu that is created *on-the-fly* and presented to the caller in a mailbox. The options on that menu are dependent on (1) the greeting that the subscriber has active, and (2) the options that the subscriber has configured.

E

E1. A digital trunking facility standard used in Europe and elsewhere, capable of transmitting and receiving 30 digitized voice or data channels. Two additional channels are used for synchronization, framing, and signalling. The transmission rate is 2048 kilobits per second. Contrast with T1.

e-business. A marketplace where businesses use Internet technologies and network computing to securely transform their business processes (using intranets), their business relationships (using extranets), and the buying and selling of goods, services, and information (using electronic commerce).

EDL. See exchange data link.

erlang. The international unit of telephony traffic, named after the Danish mathematician, Agner Krarup Erlang. The erlang has a very concise meaning for mathematicians and queuing theory experts. From a more practical standpoint, the erlang is a measure of traffic intensity, where one erlang represents one circuit occupied for one hour.

error message. Any message displayed by WebSphere Voice Response in the System Monitor as an alarm and optionally written to the WebSphere Voice Response error log, or to the AIX error log. Strictly speaking, the term *error message* should include only red (immediate

attention) and yellow (problem situation) messages, but it is also used to refer to green (a red or yellow message has been cleared) and white (informational) messages.

exchange data link. A serial connection that carries messaging information between WebSphere Voice Response and a switch.

exit. A point in a supplied application from which control can be passed to another custom-written application. On completion, the custom-written application passes control back to the supplied application. Exits are used in Unified Messaging to enable you to provide additional function.

Extensible Markup Language. The universal format for structured documents and data on the Web.

external messaging. In Unified Messaging, a system that lets subscribers send messages to, and receive messages from, subscribers on other voice mail systems.

F

File Transfer Protocol (FTP). In Transmission Control Protocol/Internet Protocol (TCP/IP), an application protocol used for transferring files to and from host computers.

formant text-to-speech. Text-to-speech that uses standard synthesized voice sounds. See also concatenative text-to-speech.

FTP. See File Transfer Protocol (FTP).

G

gateway. A component of a Voice over IP network. A gateway provides a bridge between VoIP and circuit-switched networks.

grammar files. Files listing words and phrases, that VoiceXML applications use to recognize speech.

greeting. In voice mail, the recording heard by a caller when they reach a subscriber's mailbox. See also announcement-only greeting.

greeting header. In voice mail, a recording made by a subscriber and played to callers either before or instead of a personal greeting.

H

hang up. To terminate a call. See also disconnect.

HML. See Hyper Text Mark Up Language.

hook flash. A signal sent to a switch to request a switch feature (such as call transfer).

host application. An application residing on the host computer.

HTML. See Hyper Text Mark Up Language.

HTTP. See Hyper Text Transfer Protocol.

hunt group. A set of telephone lines from which a non-busy line is hunted to handle, for example, an incoming call.

hunt group. A set of telephone lines from which a non-busy line is hunted to handle, for example, an incoming call.

Hyper Text Mark Up Language. A text based way of describing data for transmission over the internet.

Hyper Text Transfer Protocol. A protocol that defines the way in which a web server and a web client communicate with each other.

I

IETF. See Internet Engineering Task Force.

Internet Engineering Task Force. An international organization that defines and develops internet standards and specifications

IMAP4. See Internet Mail Access Protocol (IMAP4).

glossary

in-band. In the telephony voice channel, signals are said to be carried in-band.

incoming mail. In Unified Messaging, new messages recorded by callers or sent by subscribers.

input parameter. Data received by a program such as a prompt, custom server, or state table from the program that invoked it.

integrated messaging. A messaging system in which more than one copy of a single message is stored, the copies being kept synchronized by the applications used to access them. Contrast with Unified Messaging.

Integrated Services Digital Network (ISDN). A digital end-to-end telecommunication network that supports multiple services including, but not limited to, voice and data.

interactive voice response (IVR). A computer application or telephony device that uses prerecorded voice responses to provide information in response to DTMF or voice input from a telephone caller.

Internet Mail Access Protocol (IMAP4). A standard protocol for accessing mail on an e-mail server, as defined in Internet Request for Comments (RFC) 2060.

IP. See Internet Protocol .

Internet Protocol. A protocol defined by the IETF for the transmission of information over the internet

ISDN. See Integrated Services Digital Network (ISDN).

IVR. See interactive voice response (IVR).

J

Java™ and VoiceXML environment. The code that allows a Java voice application to communicate with a base WebSphere Voice Response system.

jump out. See call transfer.

K

key. One of the keys on the telephone keypad. In some contexts, the dual-tone multifrequency (DTMF) signal that corresponds to a key.

key pad. The part of the telephone that contains the push-button keys.

key pad mapping. The process of assigning special alphanumeric characters to the keys on a telephone key pad so that the telephone can be used as a computer terminal keyboard.

L

label. An optional name for a state in a state table.

LDAP. See lightweight directory access protocol (LDAP).

licensed program product (LPP). A separately-priced program and its associated materials that bear an IBM copyright and are offered under the terms and conditions of a licensing agreement.

Lightweight Directory Access Protocol (LDAP). A standard protocol for accessing directory entries on a directory server, as defined in Internet Request for Comments (RFC) 1777.

local area network (LAN). A network in which computers are connected to one another within a limited geographical area.

local node. See local system.

local system. A system that forms part of the Unified Messaging single system image (SSI). Unified Messaging sends voice messages to, and receives voice messages from, profiles on such systems, using built-in WebSphere Voice Response functions.

local variable. A user-defined temporary variable that can be accessed only by the program (state table or prompt) for which it is defined.

M

mailbox. In a voice mail system, the place where voice messages are stored.

mailbox number. An application profile with a profile ID that is usually the subscriber's extension number.

MB. Megabyte.

MCIT. See Message Center Interface Tool (MCIT).

megabyte. (1) For processor storage and real and virtual memory, 1 048 576 bytes. (2) For disk storage capacity and transmission rates, 1 000 000 bytes.

menu. A list of selectable actions available at different points in the operation of a program. In Unified Messaging the menus are spoken.

Menu-routing. A Unified Messaging voice application that allows a caller to select from up to ten destinations to which they want to transfer. This transfer is achieved by pressing a single key on the telephone keypad. If the requested number answers, the application connects the call between the caller and the number called. Otherwise, the application sends the caller to the voice mailbox for that number.

message attributes. In Unified Messaging, the priority, privacy, delivery date, and acknowledgment status of a voice message.

Message Center. An earlier name for Unified Messaging, a unified messaging application that runs on WebSphere Voice Response for AIX.

Message Center Interface Tool (MCIT). An earlier name for the Unified Messaging Interface Tool a menu-based application used to administer a Unified Messaging system.

message delivery preference. The subscriber's choice of whether their voice mail is stored as voice mail only, as e-mail only, or as both voice mail and e-mail.

message delivery type. The format in which a voice message is delivered.

message header. In Unified Messaging, information about the sender of a voice message and the date and time it was sent, which is usually played before the message itself.

message waiting indicator. A visible or audible indication (such as a light or a stutter tone) that a voice message is waiting to be retrieved.

MIME. See Multipurpose Internet Mail Extensions (MIME).

Multipurpose Internet Mail Extensions (MIME). A protocol used on the Internet for extending e-mail capability and integrating it with other forms of communication, such as voice mail and fax.

MWI. See message waiting indicator.

N

Network File System (NFS). A protocol, developed by Sun Microsystems, Incorporated, that allows any host in a network to gain access to another host or netgroup and their file directories. In a single system image (SSI), NFS is used to attach the WebSphere Voice Response DB2 database.

node code. In Unified Messaging, the numeric code that identifies remote nodes.

notification schedule. In Unified Messaging, the specification of times and phone numbers at which the subscriber is to be notified about incoming messages. The priority of messages required to trigger a notification is also specified in the schedule.

O

off-hook. A telephone line state, usually induced by lifting a receiver, in which the line is ready to make a call.

glossary

on-hook. A telephone line state, usually induced by hanging up a receiver, in which the line is ready to receive a call.

out-of-band. Within the telephony signalling channel, as opposed to the voice channel, signals are said to be carried out-of-band.

outgoing mail. In voice mail, messages sent by a subscriber to another subscriber on the same system, which have not yet been accessed by the addressee.

P

PABX. See private automatic branch exchange (PABX).

pack. A component that fits in the 9295 Multiple Digital Trunk Processor. See also RPACK, SPACK, and VPACK.

Pack Configuration. A WebSphere Voice Response for AIX tool that simplifies the process of configuring a telephony environment.

partition. A logical division of a Unified Messaging system that has its own administrator and subscribers who can be isolated from other subscribers outside that partition.

partition administrator. An administrator with permission to perform subscriber administration only for a particular partition.

password. A unique string of characters known to a computer system and to a user, who must specify the character string to gain access to the system and to the information stored in it.

PBX. See private branch exchange (PBX).

PCI. See peripheral component interconnect (PCI).

peripheral component interconnect (PCI). The rules that define how subsystems and adapters use the Intel™ bus in a computer.

personal directory. A directory of up to ten entries that certain subscribers can maintain using a Web interface. This personal directory

enables subscribers to (1) filter their remote e-mail messages based on a particular person or group of people, and (2) forward remote e-mail messages to other people listed in this directory.

personal greeting. In voice mail, a greeting recorded by a subscriber. Contrast with system greeting.

plug-in. An accessory program used to alter, enhance or extend the operation of a parent application program. Plug-ins are used in the Java and VoiceXML environment to provide text-to-speech and speech recognition services.

POP. See Post Office Protocol .

Post Office Protocol . A protocol used to retrieve e-mail from a web server.

POP3. A standard protocol for accessing mail on an e-mail server, as defined in Internet Request for Comments (RFC) 1725.

port. In time-slot management, one end of a 64 kbps unidirectional stream which can be attached to the SCBus.

port set. In time-slot management, a collection of ports which can be connected using a single CA_TDM_Connect() API call to a complementary collection of ports.

private automatic branch exchange (PABX). An automatic private switching system that services an organization and is usually located on a customer's premises. Often used synonymously with private branch exchange (PBX).

private branch exchange (PBX). A switch inside a private business that concentrates the number of inside lines into a smaller number of outside lines (trunks). Many PBXs also provide advanced voice and data communication features. Often used synonymously with private automatic branch exchange (PABX).

process a call. To answer the telephone and perform the appropriate tasks.

production system. A WebSphere Voice Response system that is used to respond to or

make *live* calls. A production system can also be used to develop new applications.

program temporary fix (PTF). An update to IBM software.

program data. Application-specific data that can be associated with a call transfer from CallPath to WebSphere Voice Response, or in the opposite direction. This is equivalent to CallPath program data, but WebSphere Voice Response imposes the restriction that the data must be a printable ASCII character string, with a maximum length of 512 bytes.

prompt. (1) A message requesting input or providing information. Prompts are seen on the computer display screen and heard over the telephone. (2) In WebSphere Voice Response, a program that uses logic to dynamically determine the voice segments to be played as a voice prompt.

prompt directory. A list of all the prompts used in a particular voice application. Used by the state table to play the requested voice prompts.

protocol. A set of semantic and syntactic rules that determines the behavior of functional units in achieving communication.

PTF. See program temporary fix (PTF)

pushbutton. (1) A key on a telephone key pad. (2) A component in a window that allows the user to invoke a specific action.

pushbutton telephone. A type of telephone that has pushbuttons. It may or may not send tone signals. If it does, each number and symbol on the key pad has its own specific tone.

Q

Quad Adapter. See Digital Trunk Quad Adapter

quick message number. In Unified Messaging, the number that callers can dial to send messages to subscribers without ringing their phone and without having to sign on to Unified Messaging.

quiesce. To shut down a channel, a trunk line, or the entire system gracefully. The shutdown is performed on a channel-by-channel basis. Channels in an idle state are shut down immediately. Channels processing calls are shut down at call completion.

R

reboot. To reset or restart the IBM System p5 or BladeCenter computer.

reduced instruction set computer (RISC). The system on which WebSphere Voice Response runs, specifically referred to as an IBM System p5 or BladeCenter computer.

referral number. The phone number to which calls are routed when call forwarding is active.

remote e-mail. E-mail stored on a separate e-mail server rather than within the Unified Messaging system. In order to be accessible by Unified Messaging, such remote e-mail servers must be Internet Mail Access Protocol (IMAP4)-compliant.

remote name. A voice file for the spoken names of profiles on a remote system.

remote node. See remote system.

remote system. Any system with which Unified Messaging can exchange voice messages, including other Unified Messaging systems and other suppliers' voice mail systems.

result. An indicator of the success or failure of a state table action, returned by WebSphere Voice Response to the state table.

result state. The state following each of the possible results of an action.

return code. A code that indicates the status of an application action when it completes.

RISC. See reduced instruction set computer (RISC).

RPACK. Resource pack. A specialized adapter card which is housed in the 9295 Multiple Digital

glossary

Trunk Processor. This adapter is used for speech recognition support. Contrast with RPACK and VPACK.

S

SCbus. See Signal Computing bus (SCbus).

segment ID number. One or more numbers used to identify a voice or prompt segment.

Session Initiation Protocol (SIP). A signaling protocol used for VoIP telephony.

Shared number. A single telephone number that hosts multiple Unified Messaging mailboxes. A caller can choose to leave a message with one of the mailboxes.

Signal Computing System Architecture (SCSA). An architecture that supports the interoperability of software and hardware components developed by different vendors in the computer telephony industry.

Signal Computing bus (SCbus). A time division multiplexed (TDM) hardware bus that interconnects different vendors' computer telephony adapters. Specified as part of Signal Computing System Architecture (SCSA).

Signal Computing System Architecture (SCSA). An architecture that supports the interoperability of software and hardware components developed by different vendors in the computer telephony industry.

signaling. The exchange of control information between functional parts of the system in a telecommunications network.

signaling process. A WebSphere Voice Response component that controls signaling for an exchange data link or common channel signaling protocol. Some signaling processes are supplied with WebSphere Voice Response, and others can be custom-written.

Signaling System Number 7 (SS7). A signaling protocol used to communicate between telephony equipment.

Simple Mail Transfer Protocol. The base TCP/IP protocol for sending and receiving e-mail.

Simple Message Desk Interface (SMDI). A messaging protocol used by Unified Messaging to send message waiting indications.

sign-on prompt. In Unified Messaging, the prompt that asks subscribers to enter their extension number and password (heard when subscribers dial Unified Messaging's number or select sign-on when listening to a greeting).

Simplified Message Service Interface (SMSI). A protocol running on a serial connection (see exchange data link) that carries messaging information between WebSphere Voice Response and Lucent or AT&T switches.

Single System Image (SSI). A cluster of WebSphere Voice Response systems that are connected together using a local area network. Each system (known as a node) in the cluster is configured as either a client or a server.

SIP. See Session Initiation Protocol (SIP).

SMDI. See Simple Message Desk Interface.

SMIT. See System Management Interface Tool (SMIT).

SMSI. See Simplified Message Service Interface (SMSI).

SMTP. See Simple Mail Transfer Protocol.

SNA. Systems Network Architecture.

SPACK. A logical component consisting of a base card, which connects to the digital trunk adapter in the pSeries[®], and a trunk interface card (TIC), which manages the trunk connection to the switch. The SPACK is required to implement common channel signalling protocols, for example, SS7. Contrast with RPACK and VPACK.

special character. A character that is not alphabetic, numeric, or blank. For example, a comma (,) or an asterisk (*).

speech synthesis. The creation of an approximation to human speech by a computer concatenating basic speech parts together. See also text-to-speech technology.

state. One step in the logical sequence of actions that comprises a WebSphere Voice Response voice application.

state table. A list of all the actions used in a particular voice application. A component of WebSphere Voice Response.

state table action. One instruction in a set of instructions contained in a WebSphere Voice Response state table that controls how WebSphere Voice Response processes various operations such as playing voice prompts or recording voice messages. See also state.

SS7. See Signaling System Number 7 (SS7).

SSI. See Single System Image (SSI).

subscriber. In voice mail, any person who owns a mailbox. Contrast with caller.

subscriber class. A named set of variables used to define a specific level of service available to telephone subscribers, such as maximum number of messages per mailbox and maximum number of members per mailbox distribution list.

subscriber type. A setting for a subscriber that determines which menu options and features are available to that subscriber and to callers into that subscriber's mailbox. There are a Standard set of menus and options, and an additional four subscriber types: Business - local & remote, Business - local, Remote e-mail only, and Residential.

super administrator. An administrator who can create and delete partitions. This administrator can also perform subscriber administrator for any partition.

switch. A generic term used to describe a telecommunications system that provides connections between telephone lines and trunks.

system administrator. The person who controls and manages the WebSphere Voice Response system by adding users, assigning account numbers, and changing authorizations.

system greeting. In voice mail, a default greeting heard by callers to the mailboxes of subscribers who have not recorded a personal greeting or who have selected the system greeting. Contrast with personal greeting.

System Management Interface Tool (SMIT). A set of utilities that can be used for various purposes, such as loading WebSphere Voice Response software, installing the exchange data link, and defining SNA profiles.

system parameter. A variable that controls some the behavior of WebSphere Voice Response or applications running under WebSphere Voice Response. System parameters are set using System Configuration or Pack Configuration options on the Configuration menu. Some system parameter values are assigned to system variables when an application is initialized.

system variable. A permanent global variable defined by WebSphere Voice Response for use by state tables. Many system variables are loaded with values when the state table is initialized. Some values are taken from system parameters.

T

T1. A digital trunking facility standard used in the United States and elsewhere, capable of transmitting and receiving 24 digitized voice or data channels. The transmission rate is 1544 Kilobits per second. Contrast with E1.

tag image file format/fax (TIFF/F). A graphic file format used to store and exchange scanned fax images.

TCP. See Transport Control Protocol.

TCP/IP. See Transmission Control Protocol/Internet Protocol (TCP/IP).

telephone keypad interface. The original interface for Unified Messaging. Subscribers dial

glossary

in from a telephone that generates tones, and select menu choices using the telephone's keypad.

telephony portal. The use of Unified Messaging solely for telephony. All messages and required subscriber information is retrieved from external e-mail and LDAP servers.

text-to-speech (TTS). The process by which ASCII text data is converted into synthesized speech. See also speech synthesis.

TIFF/E. See tag image file format/fax (TIFF/F).

time slot. The smallest switchable data unit on a data bus, consisting of eight consecutive bits of data. One time slot is equivalent to a data path with a bandwidth of 64 Kbps.

token-ring network. A local area network that connects devices in a ring topology and allows unidirectional data transmission between devices by a token-passing procedure. A device must receive a token before it can transmit data.

tone. An audible signal sent across a telephone network. There are single (one-frequency) tones, tritones (three sequential tones at different frequencies), dual tones (two simultaneous tones at different frequencies), and dual sequential tones. Each has a different meaning..

transaction. A specific, related set of tasks within an application that retrieve information from a file or database. For example, a request for the account balance or the available credit limit.

transaction messaging. The ability to associate an item of data, such as a transaction identifier, with a voice message. The voice message can subsequently be retrieved by referencing the data value.

transfer. See call transfer.

Transmission Control Protocol/Internet Protocol (TCP/IP). A communication subsystem that is used to establish local area and wide area networks.

Transport Control Protocol. A protocol defined by the IETF, used for communicating across the internet

trombone. A connected voice path that enters an IVR from a switch on one circuit, then returns to the same switch on a parallel circuit. Two IVR ports and two circuits are used, but in some circumstances this might be the only way to make a connection between two callers if the attached switch does not support a Call Transfer function. Also known as double-trunking.

trunk. A telephone connection between two central offices or switching devices. In WebSphere Voice Response, a trunk refers to 24 or 30 channels carried on the same T1 or E1 digital interface.

TTS. See text-to-speech technology.

trunk interface card (TIC). The component of the VPACK that manages the trunk connection to the switch.

U

unified messaging. A messaging system in which a single copy of a message is stored and accessed by multiple applications (for example, voice mail and e-mail). Contrast with integrated messaging.

Unified Messaging Interface Tool. A menu-based application used to administer a Unified Messaging system.

user. Someone who uses Unified Messaging as a system administrator, application developer, or similar. Contrast with caller.

V

variable. A system or user-defined element that contains data values used by WebSphere Voice Response voice applications.

voice application. A WebSphere Voice Response application that answers or makes calls, plays recorded voice segments to callers, and responds to the caller's input.

voice bridge. A device that connects a telephone switch to WebSphere Voice Response.

voice directory. A list of voice segments identified by a voice directory name. Voice directories can be referenced by prompts and state tables.

Voice Interface. A VoiceXML application that allows voice access to Unified Messaging. By dialling-in from any telephone, a subscriber can use spoken commands to access, process, and send voice mail, fax, and e-mail messages.

voice mail. The capability to record, play back, distribute, and route voice messages.

voice mailbox. The repository for incoming messages for a voice mail subscriber. It may also contain data about how incoming calls or messages are to be handled. For example, it may identify the ReachMe number, notification schedules, e-mail address to which messages are to be sent, and so on.

voice message. In Unified Messaging, a recording made by a caller for later retrieval by a subscriber.

voice messaging. The capability to record, play back, distribute, route, and manage voice recordings of telephone calls through the use of a processor, without the intervention of agents other than the callers and the message recipients.

Voice Protocol for Internet Messaging (VPIM). Standard for digital exchange of voice messages between different voice mail systems, as defined in Internet Request For Comments (RFC) 1911.

voice recognition. The capability of a computer to understand the spoken word for the purpose of receiving commands and data input from the speaker.

voice response beans. Java beans with which you can build voice applications.

voice response node. A node where a base WebSphere Voice Response system is running, providing access to voice processing resources, and optionally running applications.

voice segment. The spoken words or sounds that comprise recorded voice prompts. Each segment in an application is identified by a language, a voice directory name, and a segment ID and usually includes accompanying text.

voice table. A grouping of voice segments in a table for access using an index, such as the numbers 0 to 9 or the letters A to Z. Voice tables can be referenced by prompts, but not directly by state tables.

VoIP. See Voice over IP

Voice over IP. The sending of telephony voice over Internet Protocol (IP) data connections instead of over existing dedicated voice networks, switching and transmission equipment. See gateway.

VPACK. A component consisting of a base card, which connects to the digital trunk adapter in the pSeries, and a trunk interface card (TIC), which manages the trunk connection to the switch. The single digital trunk processor contains one VPACK, and the multiple digital trunk processor contains slots for up to five VPACKs. Contrast with RPACK and SPACK.

VPIM. See Voice Protocol for Internet Messaging (VPIM).

W

WAP. See Wireless Application Protocol.

WebSphere Application Server. An IBM Web application server. It provides software to deploy, integrate and manage business applications, including handling transactions and extending back-end business data and applications to the Web.

WebSphere Voice Response. A voice processing system, bringing together telephone and data communications networks to use information stored in databases directly from a telephone.

WebSphere Voice Server. An IBM voice middleware product that enables developers to create and deploy voice applications. It includes

glossary

speech recognition and text-to-speech engines, development tools, and a telephony platform connector.

Web interface. An interface for Unified Messaging. Subscribers access and manage their messages using a Web browser.

Wireless Application Protocol. An open specification based on XMP and IP that enables standard data formats and transmissions for wireless devices.

Wireless Markup Language. A subset of “Extensible Markup Language” on page 51, developed for “Wireless Application Protocol.”

window. An area of the screen with defined borders in which information is displayed. A window may be equal in size to the entire screen, or may share the screen with other windows.

WML. See Wireless Markup Language.

X

XML. See Extensible Markup Language.

Product documentation

This section contains a list of the documentation for Unified Messaging, WebSphere Voice Response for AIX and associated products. Hardcopy books, where available, can be ordered through your IBM representative or through the IBM Publications Center at <http://www.ibm.com/shop/publications/order>. At this Web site you can also obtain PDF and HTML versions of the documentation.

Unified Messaging publications can also be found by going to the IBM Pervasive software Web site at <http://www.ibm.com/software/pervasive>, selecting the **products** link, and then selecting the **library** link from the WebSphere Voice Response page.

PDF and HTML versions of the Unified Messaging publications are available on the CD-ROM supplied with the product (order number SK2T-1787). Only the *IBM Unified Messaging for WebSphere Voice Response: General Information and Planning* book is available as a printed book. In addition, WebSphere Voice Response for AIX and WebSphere Voice Response for Windows publications are available together in PDF and HTML formats on a separate CD-ROM.

- *IBM Unified Messaging for WebSphere Voice Response: General Information and Planning*, GC34-6398
- *IBM Unified Messaging for WebSphere Voice Response: Administrator's Guide*, SC34-6399
- *IBM Unified Messaging for WebSphere Voice Response: Subscriber's Guide (Types 0,1,2,3,4 and 9)*, SC34-6403
- *IBM Unified Messaging for WebSphere Voice Response: Subscriber's Guide (Types 5,6,7 and 8)*, SC34-6400
- *IBM Unified Messaging for WebSphere Voice Response: Voice Interface*, SC34-6401

Note: To read PDF versions of books you need to have the Adobe Acrobat Reader (it can also be installed as a plug-in to a Web browser). It is available from Adobe Systems at <http://www.adobe.com>.

WebSphere Voice Response base software

- *WebSphere Voice Response for AIX: General Information and Planning*, GC34-6379
- *WebSphere Voice Response for AIX: Installation*, GC34-6380
- *WebSphere Voice Response for AIX: User Interface Guide*, SC34-6386

- *WebSphere Voice Response for AIX: Configuring the System*, SC34-6381
- *WebSphere Voice Response for AIX: Managing and Monitoring the System*, SC34-6384
- *WebSphere Voice Response for AIX: Designing and Managing State Table Applications*, SC34-6388
- *WebSphere Voice Response for AIX: Application Development using State Tables*, SC34-6387
- *WebSphere Voice Response: Developing Java applications*, GC34-6377
- *WebSphere Voice Response for AIX: Deploying and Managing VoiceXML and Java Applications*, GC34-6378
- *WebSphere Voice Response for AIX: Custom Servers*, SC34-6389
- *WebSphere Voice Response for AIX: 3270 Servers*, SC343-6390
- *WebSphere Voice Response for AIX: Problem Determination*, GC34-6382
- *WebSphere Voice Response for AIX: Fax using Brooktrout*, GC34-6385
- *WebSphere Voice Response for AIX: Cisco ICM Interface User's Guide*, SC34-6391
- *WebSphere Voice Response for AIX: Programming for the ADSI Feature*, SC34-6392
- *WebSphere Voice Response for AIX: Programming for the Signaling Interface*, SC34-6392
- *WebSphere Voice Response for AIX: Voice over IP using Session Initiation Protocol*, GC34-6383
- *WebSphere Voice Response for AIX: Using the CCXML Browser*, GC34-6368

WebSphere Voice Response related products

WebSphere Voice Server; Use with IBM WebSphere Voice Response for AIX

- *WebSphere Voice Server for AIX: Administrator's Guide*, G210-1561
- *WebSphere Voice Server for AIX: Application Development using State Tables*, G210-1562

The WebSphere Voice Server Version 4.2 documentation can be downloaded free of charge from the IBM Publications Center at <http://www.ibm.com/shop/publications/order>.

WebSphere Voice Server SDK V1.5 Programmer's Guide is available in softcopy by going to the WebSphere Voice Server Web page at <http://www.ibm.com/software/webservers/voiceserver/support.html> and clicking on **Product information**.

AIX

- *AIX: Operating System Installation: Getting Started*, SC23-4388
- *AIX: Installation Guide and Reference*, SC23-4389
- *AIX: System User's Guide; Operating System and Devices*, SC23-4121
- *AIX: System User's Guide; Communications and Networks*, SC23-4122
- *AIX: System Management Guide; Operating System and Devices*, SC23-4126
- *AIX: System Management Guide; Communications and Networks*, SC23-4127
- *AIX: Security Guide*, SC23-4860

Computers running AIX

- *RS/6000 and pSeries: Site Hardware and Planning Information*, SA38-0508
- *System p5: IBM Systems Hardware Information Center*. Information on physical site planning and preparation is provided in the information center: <http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/index.jsp>
- *BladeCenter: BladeCenter Information Center*. Information on physical site planning and preparation is available through the information center: <http://www-03.ibm.com/systems/bladecenter/>

Brooktrout Fax

The following publications provide further information on Brooktrout Fax

- *Brooktrout Technology: TR114 PCI Digital Hardware Guide*, 930-412-72, Jan 1999
- *Brooktrout Technology: Brooktrout Fax and Voice API: User's Guide, Software Version 4.3*, 930-063-10, March 2000
- *Brooktrout Technology: Brooktrout Fax and Voice Application Programming Interface: Programmer's Reference Manual Version 4.3*, 930-064-10, March 2000
- *Brooktrout Technology: Bfo API and Driver Installation Guide for TR114, Software Version 4.3*, 930-455-00, July 2000

Index

A

accessibility x
Axis WSDL2Ws 4

B

business integration 1

C

call forward number, get 18
call forward number, set 18
change password 32
code generation tools
 Axis WSDL2Ws 4
 IBM Rational Application Developer 4
 WSDL 4
 Wsd.exe 4

D

delete message 26

E

email notification, get 20
email notification, set 19
examples
 SOAP message 2

F

folder counts, get 21
forward message 26

G

get call forward number 18
get email notification 20
get folder counts 21
get mailbox presence status 24
get message information 30
get preferences 34
get subscriber information 37
glossary 47

I

IBM Rational Application Developer 4
index 65
information
 Unified Messaging xi

L

logon
 change password 32
 validate password 32

M

mail box settings 18
 get call forward number 18
 get email notification 20
 get mailbox presence status 24
 set call forward number 18
 set email notification 19
 set mailbox presence status 22
mailbox presence status, get 24
mailbox presence status, set 22
message information, get 30
message types
 SOAP 3
message, delete 26
message, forward 26
message, save 29
messages 30
 delete message 26
 forward message 26
 get folder counts 21
 get message information 30
 save message 29
 SOAP 1
Microsoft Wsd.exe 4

P

password, change 32
password, validate 32
preferences, get 34
preferences, set 33

S

save message 29
set call forward number 18
set email notification 19
set mailbox presence status 22
set preferences 33
set subscriber information 35
SOAP 1
 example messages 2
 message types 3
standards 1
subscriber information, get 37
subscriber information, set 35
subscriber preferences 33
 get preferences 34
 get subscriber information 37
 set preferences 33
 set subscriber information 35

T

Trademarks 44

V

validate password 32

W

web services

 introduction to 1

Web Services Description Language 3

WSDL 3

 code generation tools 4

 file structure 4

Wsd.exe 4

WSDL2Ws 4



Program Number: 5724-I07

SC34-6975-00



Spine information:



Unified Messaging for
WebSphere Voice Response

Web Services Voicemail API

Version 4.2