



Adding Telephony Support to IBM Lotus Instant Messaging and Web Conferencing

Copyright and Trademark Information

Disclaimer

THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS DOCUMENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS DOCUMENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS DOCUMENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.

Licensed Materials - Property of IBM

©Copyright IBM Corporation 1998, 2004 All rights reserved.

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

Lotus Software

IBM Software Group

One Rogers Street

Cambridge, MA 02142

List of Trademarks

IBM, the IBM logo, 1-2-3, AIX, AS/400, DB2, Domino, Domino Designer, iNotes, iSeries, Lotus, Lotus Notes, MQSeries, Netfinity, Notes, QuickPlace, Sametime, SmartSuite, S/390, Tivoli, WebSphere, and Word Pro are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Pentium is a trademark of Intel Corporation in the United States, other countries, or both.

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

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

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

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

Third Party Notices

For the XSL and XML Parser and Processor

The Apache Software License, Version 1.1

Copyright (c) 1999-2000 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
4. The names "Xerces" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.
5. Products derived from this software may not be called "Apache," nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation and was originally based on software copyright (c) 1999, International Business Machines, Inc., <http://www.ibm.com>. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

For DSIG base64

COPYRIGHT 1995 BY: MASSACHUSETTS INSTITUTE OF TECHNOLOGY (MIT), INRIA

This W3C software is being provided by the copyright holders under the following license. By obtaining, using and/or copying this software, you agree that you have read, understood, and will comply with the following terms and conditions:

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee or royalty is hereby granted, provided that the full text of this NOTICE appears on ALL copies of the software and documentation or portions thereof, including modifications, that you make.

THIS SOFTWARE IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS

OR OTHER RIGHTS. COPYRIGHT HOLDERS WILL BEAR NO LIABILITY FOR ANY USE OF THIS SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

For STLport

License Agreement

Boris Fomitchev grants Licensee a non-exclusive, non-transferable, royalty-free license to use STLport and its documentation without fee.

By downloading, using, or copying STLport or any portion thereof, Licensee agrees to abide by the intellectual property laws and all other applicable laws of the United States of America, and to all of the terms and conditions of this Agreement.

Licensee shall maintain the following copyright and permission notices on STLport sources and its documentation unchanged :

Copyright 1999,2000 Boris Fomitchev

This material is provided "as is", with absolutely no warranty expressed or implied. Any use is at your own risk.

Permission to use or copy this software for any purpose is hereby granted without fee, provided the above notices are retained on all copies. Permission to modify the code and to distribute modified code is granted, provided the above notices are retained, and a notice that the code was modified is included with the above copyright notice.

The Licensee may distribute binaries compiled with STLport (whether original or modified) without any royalties or restrictions.

The Licensee may distribute original or modified STLport sources, provided that:

The conditions indicated in the above permission notice are met; The following copyright notices are retained when present, and conditions provided in accompanying permission notices are met :

Copyright 1994 Hewlett-Packard Company

Copyright 1996,97 Silicon Graphics Computer Systems, Inc.

Copyright 1997 Moscow Center for SPARC Technology.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Hewlett-Packard Company makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Silicon Graphics makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Moscow Center for SPARC Technology makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Copyright 2001 by STLport

For MD5 hash

Copyright (C) 1990, RSA Data Security, Inc. All rights reserved. License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

For Log4J Logging

The Apache Software License, Version 1.1 at <http://www.apache.org/LICENSE>, 24 May 2002

The Apache Software License, Version 1.1

Copyright (c) 2000 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

6. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
7. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
8. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
9. The names "Apache" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.
10. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

Portions of this software are based upon public domain software originally written at the National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign.

Contents

Chapter 1	About This Paper	1
Chapter 2	Overview: Enabling Telephony Support	2
Chapter 3	Creating the Server Application.....	4
Chapter 4	Enabling Clients	10
	Sametime Clients	10
	Web Pages	10
Appendix A	XML String Format and Creation	12

Chapter 1 About This Paper

Intended Audience

This white paper is for telephony providers who want to add telephony support for users of the Lotus® Sametime® Connect client and Sametime-enabled Web pages, to allow online users to initiate telephone calls to selected community members.

Note Throughout this white paper, the IBM® Lotus Instant Messaging and Web Conferencing (Lotus Sametime) product is referred to as “Sametime,” and its associated toolkits are referred to as “Sametime” toolkits.

This paper does not include information about Sametime programming. For more information on Sametime programming, see the Sametime Software Development Kit documentation.

Related Documents

- Sametime Community Server Toolkit Developer’s Guide
- Sametime Community Server Toolkit Tutorial
- Sametime Java Toolkit Developer’s Guide
- Sametime Java Toolkit Tutorial
- Sametime Links Toolkit Developer’s Guide and Reference

Requirements

The Sametime Community Server Toolkit can be used in any JDK 1.1 or higher Java development environment.

The toolkit is targeted for use with the following Sametime servers:

- Windows – 2.5 and above.
- iSeries – 3.1 and above

Although applications developed with this toolkit will work when run on a Sametime 2.x or later server, toolkit services that require features new to this release will not function. In particular, the code examples in the toolkit should be run on the latest version of the Sametime server.

Users of the following may initiate telephone calls:

- Sametime 3.1 and later Connect clients
- custom Sametime clients created with the Sametime Java Toolkit (version 3.1 and later)
- Web pages with links enabled via the Sametime Links Toolkit (version 3.1 and later)

Users of older clients and Web pages cannot initiate calls, but may participate as invitees.

Additional Information

Additional information about Sametime can be found at the following Web sites:

- <http://www.lotus.com/sametime>
- <http://www.ibm.com/developerworks/lotus>
- <http://www.redbooks.ibm.com/>

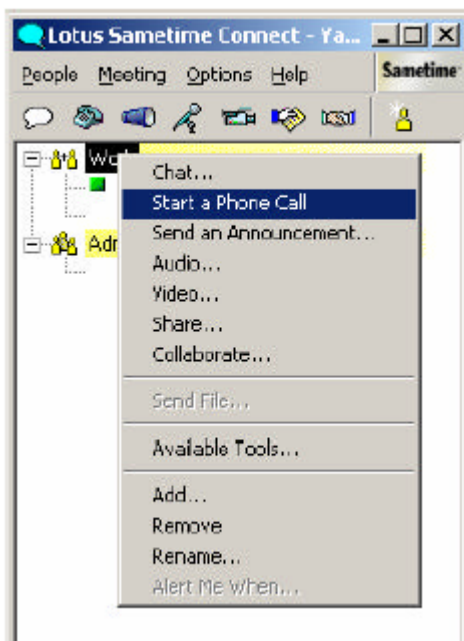
Chapter 2 Overview: Enabling Telephony Support

Telephony support in Sametime is enabled with a server application that you create, using the Sametime Community Server Toolkit. This server application contains your telephony implementation, and also sends a server attribute to indicate that telephony support is present on the server.

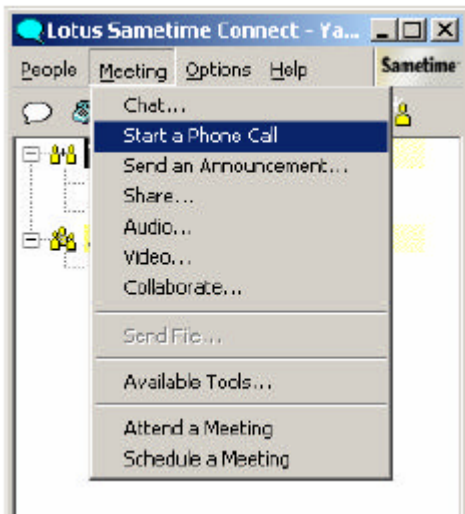
For users of Sametime Connect clients and custom Sametime clients created with the Sametime Java Toolkit, no further work is needed. If your users are accessing Sametime-enabled Web pages, you must modify the JavaScript in those pages by adding a call to a new function, `STLinksStartCall`, where needed.

When telephony support is enabled:

- Users of Sametime-enabled Web pages can specify user names and initiate a call to those users, via input devices (such as text fields and buttons) that you provide.
- Users of Sametime clients can right-click an active user name or a group name to display a new “Start a Phone Call” shortcut menu item and immediately start a call to the active user or all active users in the group.



- A new telephone button (visible in the toolbar in the previous graphic, and partially visible below) appears in the Sametime Connect client, and a new “Start a phone call” menu item appears in the Meeting menu:



Client users can highlight multiple names in any awareness list and use the new Meeting menu item or toolbar button to initiate a call to multiple users.

Note User names may be short or hierarchical, but must be unique when resolved. This solution does not handle name conflicts. Any conflicting names will not be included in the list of names provided to the server application.

Chapter 3 Creating the Server Application

You initiate telephone calls through a Java Sametime server application. The exact implementation is up to you; the only Sametime requirements are that you:

- create and set a server attribute to indicate that the service is active
- use the Sametime Instant Messaging Service to receive call-related messages from users

The server attribute ID is STS_ATTR_TELEPHONY_PLUG_IN (9060). The attribute value is the server application's login ID, to allow users to respond to the component. The IM type for the Instant Messaging Service is STS_IM_TYPE_TELEPHONY_PLUG_IN (9060).

When a user has selected names for a call, an XML string is generated and returned to the server application in an instant message. Your application uses the Instant Messaging Service to receive the message, and can then process the XML string in the message as needed to initiate the call. (For more information about the the XML string, see Appendix A.)

Sample

The following code is a skeletal server application with essential calls and stubs for methods you may need to process call requests.

```
import com.lotus.sametime.core.comparch.*;
import com.lotus.sametime.community.*;
import com.lotus.sametime.core.types.*;
import com.lotus.sametime.core.util.connection.*;
import com.lotus.sametime.generalawareness.*;
import com.lotus.sametime.im.*;

import java.util.Vector;
import java.util.Enumeration;
import java.util.Random;

/**
 * TelephonySA: The server application
 */
public class TelephonySA
    implements      LoginListener,
                  GeneralAwarenessListener,
                  ImServiceListener,
                  ImListener
{
    // The new server application attribute
    private final static int STS_ATTR_TELEPHONY_PLUG_IN = 9060;

    // The new instant message type
    private final static int STS_IM_TYPE_TELEPHONY_PLUG_IN = 9060;
```

```

// The session object
private STSession m_session;
// The server application service
private ServerAppService m_saService;

// The general awareness service
private GeneralAwarenessService m_gaService;

// The login ID
private String m_saLoginID;

/**
 * Initialize all variables
 */
private void initialize(String serverName)
{
    // Initialize the Sametime session and objects.
    try
    {
        // First, we create a new session, that belongs
        // uniquely to us.
        m_session = new STSession(serverName);

        // Load components
        m_session.loadSemanticComponents();
        m_saService = (ServerAppService) m_session.getCompApi(
            ServerAppService.COMP_NAME);
        m_gaService = new GeneralAwarenessComp(m_session);

        // start the session
        m_session.start();
    }

    catch (DuplicateObjectException e)
    {
        log("STSession or Components created twice.");
        e.printStackTrace();
        System.exit(1);
    }

    // Now we can login.
    login(serverName);
}

/**
 * login to Sametime as server application
 */
private void login(String serverName)
{
    // Add ourselves as login listener to the community
    m_saService.addLoginListener(this);
}

```

```

// Server applications login directly to the server, and
// not through the mux, so we can't use the default port.
Connection[] connections = {
    new SocketConnection(1516, 17000),
};

m_saService.setConnectivity(connections);
short loginType = STUserInstance.LT_SERVER_APP;
m_saService.loginAsServerApp( serverName, loginType,
    "TelephonySA", null);
}

/**
 * Implement LoginListener.loggedIn.
 */
public void loggedIn(LoginEvent event)
{
    // Save the login ID to use as the value of the new
    // server attribute.
    Login login = event.getLogin();
    STUserInstance instance = login.getMyUserInstance();
    STLoginId id = instance.getLoginId();
    m_saLoginID = id.getId();

    //Start the Instant Messaging Service.
    InstantMessagingService m_imService =
        (InstantMessagingService) m_session.getCompApi
        (InstantMessagingService.COMP_NAME);
    m_imService.registerImType(STS_IM_TYPE_TELEPHONY_PLUG_IN);
    m_imService.addImServiceListener(this);

    m_gaService.addGeneralAwarenessListener(this);
}

/**
 * Implement LoginListener.loggedOut.
 */
public void loggedOut(LoginEvent event)
{
    m_imService.removeImServiceListener(this);
    m_gaService.removeGeneralAwarenessListener(this);
    // Add application-specific processing here if needed.
}

```

```

/**
 * Implement GeneralAwarenessServiceListener.serviceAvailable.
 * Create the required server attribute with the value of the
 * server application login ID, and set the server attribute.
 */
public void serviceAvailable(GeneralAwarenessServiceEvent
    event)
{
    STAttribute attr = new
        STAttribute(STS_ATTR_TELEPHONY_PLUG_IN, m_saLoginID);
    m_gaService.changeAttr(null, attr);
}

/**
 * Implement
 * GeneralAwarenessServiceListener.serviceUnavailable.
 */
public void serviceUnavailable(GeneralAwarenessServiceEvent
    event)
{
    // Add application-specific processing here if needed.
}

public void attrChanged(AttributeEvent evt)
{
    // Add application-specific processing here if needed.
}

public void changeAttrFailed(AttributeEvent evt)
{
    // Add application-specific processing here if needed.
}

public void attrRemoved(AttributeEvent evt)
{
    // Add application-specific processing here if needed.
}

public void removeAttrFailed(AttributeEvent evt)
{
    // Add application-specific processing here if needed.
}

```

```

/**
 * Handle IM events so you can process XML strings sent
 * by clients or Web pages.
 */
public void imReceived(ImEvent evt)
{
    Im im = evt.getIm();
    If (im.getType() == STS_IM_TYPE_TELEPHONY_PLUG_IN)
        im.addImListener(this);
}

public void imOpened(ImEvent evt)
{
    // Add application-specific processing here if needed.
}

public void openImFailed(ImEvent evt)
{
    // Add application-specific processing here if needed.
}

public void imClosed(ImEvent evt)
{
    // Add application-specific processing here if needed.
}

/**
 * Handle text sent via the specified IM session.
 */
public void textReceived(ImEvent evt)
{
    // Add application-specific processing here as needed.

    // Now we can close the IM.
    Im im = evt.getIm();
    im.close(0);
}

/**
 * Handle data sent via the specified IM session.
 */
public void dataReceived(ImEvent evt)
{
    // Add application-specific processing here if needed.
}

```

```
/**
 * Entry point of the application
 */
public static void main(String[] args)
{
    if ( args.length != 1 )
    {
        System.out.println("Usage: TelephonySA serverName");
        System.exit(0);
    }

    new TelephonySA().initialize(args[0]);
}
}
```

Chapter 4 Enabling Clients

Sametime Clients

Once you have created the telephony server application, Sametime Connect clients and custom Sametime Java Toolkit clients are automatically enabled. The new menu items and toolbar button shown in Chapter 2 appear and can be used to initiate telephone calls.

Web Pages

Sametime-enabled Web pages require a little more work. With telephony support active on the Sametime server, you must also add code to your Sametime-enabled Web pages to collect names and send them to the server to initiate telephone calls.

To send the user information to the server, call the following function:

STLinksStartCall(usernames, resolve)

Parameters:

usernames - The list of unique invitee names, separated by semicolons.

resolve - 1 if the usernames list contains names that must be resolved, 0 otherwise.

Usage

This function takes a list of selected users, creates the required XML string, and sends it to the telephony server application.

Sample

This HTML form collects users names and sends them to the server. Note the “resolveCallNames” checkbox. The user can uncheck this box if the names are specified in hierarchical format.

The screenshot shows a web form titled "Sametime CallStarter". It contains the following elements:

- A "Login name:" label followed by a text input field.
- A "Password:" label followed by a text input field.
- Two buttons: "Login" and "Logout".
- A "People to call:" label followed by a text input field.
- A checkbox labeled "Resolve" which is checked.
- A "Start Call" button.

The following fragment shows the code for the form:

```
<form name="f1">
...
People to call:&nbsp;<input type="text" size="30" name="callNames"
style="width:250">
<input type="checkbox" size="20" name="resolveCallNames"
checked>Resolve
<br>
<input type=button value="Start Call"
onClick=STLinksStartCall(document.f1.callNames.value,document.f1.
resolveCallNames.checked) id=button1 name=button1>
...
</form>
```

Appendix A XML String Format and Creation

When a user selects names in a Sametime client or Sametime-enabled Web page, the names of the selected users are sent to the server in an XML string, so that your server application can initiate the call. The XML has the following form:

```
<call_request_info>
<user>
<user_id> CN=firstName lastName/O=IBM</user_id>
<user_on_mobile_device>1</user_on_mobile_device>
<external_user>0</external_user>
</ user >
</call_request_info >
```

In the XML string, the lines are separated by newline characters ("`\n`").

In this example, the user is on a mobile device and belongs to the current community (that is, the user is not an external user).

In the case of the Sametime Connect client, the client creates the XML. For a Sametime Java Toolkit client, the Java Toolkit creates the XML. In the case of a Sametime-enabled Web page, it is the STLinks server application that creates the XML.