



Instant Technologies

Advanced Sametime Links Tutorial

First Published: March 16, 2003

Last revision July 8, 2003

Information in this document is subject to change without notice.
© 2003 Carl Tyler and Instant Technologies. All rights reserved
Reproduction in any manner whatsoever without written permission of Carl Tyler
is strictly forbidden. Carl Tyler and Instant Technologies shall not be liable for
errors contained herein or for incidental or consequential damages in connection
with this document.

Table of Contents

Chapter 1 - Introduction	5
Before we get started	5
Skills and knowledge required	5
The purpose of this document	5
Chapter 2 - What is ST Links?	6
Size is important	6
Limitations	6
Chapter 3 - Sametime Links directory overview	8
Directory Contents	9
Chapter 4 - Understanding the HTML files	11
Files within the "allLang" sub-directory.	11
<i>borderFrame.html</i>	11
<i>statusFrame.html</i>	11
<i>transcript.html</i>	12
Files within the individual language sub-directories.	12
<i>announceBtn.html</i>	13
<i>announceFrame.html</i>	14
<i>announcement.html</i>	14
<i>changeStatus.html</i>	15
<i>chatApplet.html</i>	15
<i>chatBtn.html</i>	16
<i>chatWindow.html</i>	16
<i>dirApplet.html</i>	17
<i>directory.html</i>	17
<i>im.html</i>	18
<i>inputFrame.html</i>	19
<i>invitation.html</i>	20
<i>invitees.html</i>	21
<i>inviteFrame.html</i>	22
<i>inviteOthers.html</i>	23
<i>meetingTools.html</i>	24
<i>nway.html</i>	25
<i>peopleHeader.html</i>	26
<i>peoplelist.html</i>	27
<i>place.html</i>	28
<i>placeChat.html</i>	28
<i>placeHeader.html</i>	28
<i>resolve.html</i>	29
<i>res.js</i>	29
Chapter 5 - Building an interactive website	30
Provide the customer with a customised/branded Sametime Links experience	30
<i>Display a company logo in the top of the chat window</i>	30

<i>Remove the buttons to “Invite Others...” and “Add Tools...”</i>	31
Allow an agent to see who is on their web site	32
<i>Modify the website pages to load the Sametime Links applet</i>	33
<i>Log the visitor into Sametime</i>	33
<i>Upon a successful login, move the customer in to the place</i>	
“WEBSITE”	33
<i>Provide a web page for the agent to login and watch visitors</i>	34
<i>Display on the agents page the number of people in the</i>	
“WEBSITE” place	35
Allow the agent to see which page the customer is on	36
<i>Let other agents know when an agent is already chatting with a</i>	
customer	37
Chapter 6 - Conclusion	39
About the author	39
About Instant Technologies	39

Chapter 1 - Introduction

The purpose of this document it is to expand on the documentation that Lotus has created for Sametime Links to give an understanding of how Sametime Links may be used in a Internet/Intranet environment to provide proactive interaction with visitors. I have the Sametime Links documentation to be factually correct (in the most part), but it's usefulness could be expanded greatly if it included more examples; the same could also be said for the other Sametime toolkits and their documentation, with more examples things would become much clearer; in a future Sametime, I would love to see an example for every function that is available, much in the way that every function in the Lotus Domino Designer Help has an example. In the mean time I hope this document will stop others going through the pain I did trying to figure out how Sametime Links works.

Before we get started

A few things to note:

1. There will very probably be mistakes, if you find one, e-mail me at ctyler@instant-tech.com and I'll correct them.
2. I can answer some questions, but keep in mind I am running a business, Instant Technologies builds Instant Messaging solutions and offers consulting services, so I will offer help where I can, but I cannot offer help if it impacts the running of Instant Technologies; sorry to sound harsh, but our employees have to eat.
3. Contents of this document are © 2003 Carl Tyler and Instant Technologies.

Skills and knowledge required

To understand the contents of this document, you're going to need some skills in a few web based technologies, which are Javascript and HTML, I've tried to write this document to explain most things, but sometimes I assume that everyone else knows at least what I know, sometimes can skim over details, if you have a good understanding of Javascript and HTML, you should have no problems.

The purpose of this document

Using this document we are going to build an application that adds Sametime awareness to a web site using Sametime Links. Our sample application has some simple goals, which will enable you to learn more about Sametime Links while allowing you to add awareness to a web site. The goals for this example are:

1. Provide the visitor with a customised/branded Sametime Links experience
2. Allow an agent to see who is on their web site
3. Allow the agent to see which page the visitor is on
4. Allow the agent to interact with the visitor
5. Let other agents know when a visitor is already chatting with a visitor

So now that we have covered most of the basics, and you're still reading, lets move on...

Chapter 2 - What is ST Links?

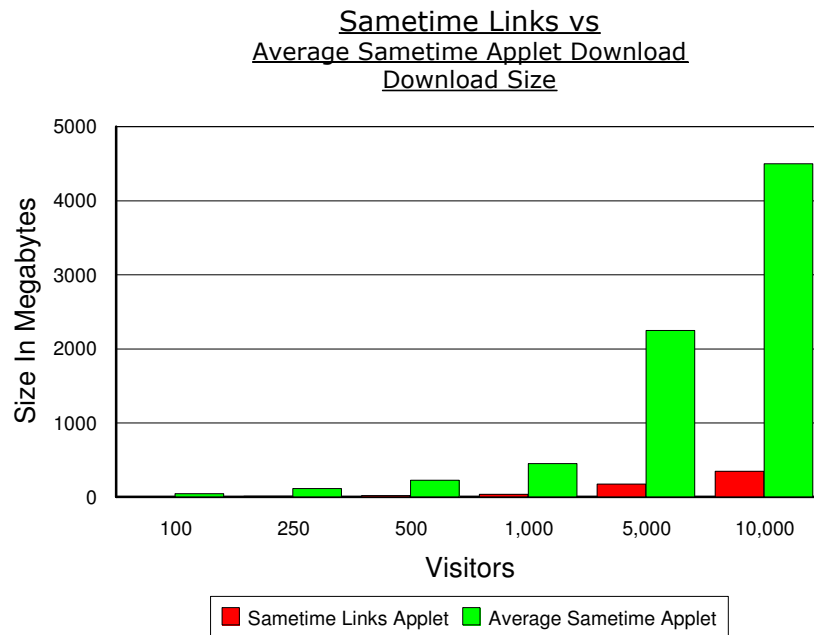
A good place to start with getting a better understanding of Sametime Links is to read how IBM/Lotus describes it, they state:

Sametime Links is a light toolkit that allows Web developers to Sametime-enable their web pages and applications with "live names." A simple HTML/JavaScript API allows web developers to turn existing names into Sametime links by simply adding a few lines of HTML, without affecting the layout of the page. While rich in functionality, it is light in size, using an embedded applet of only about 20K. - IBM/Lotus Sametime Links Release Notes

I think that's a pretty good summary of Sametime Links, and it's basically true, although the final shipping Sametime Links applet is around 25K and 35K depending on JVM, but what's 15K between friends.

Size is important

Sametime Links tends to be much smaller than other Sametime web applications that have been built to provide this kind of solution, other solutions often make use of the Java Sametime Community toolkit to build this type of app (this is how I've done it in the past) , and through lazy programming or lack of knowledge they often include the whole of the STcomm.JAR and STRes.jar files, which results in a download of about 450K, so it's easy to see the benefits of Sametime Links from a size point of view. This simple chart shows how that difference can impact your web site network traffic for a single day.



This really has the biggest impact on visitors to your website, the first time someone visits a page using Sametime Links it loads roughly 12x quicker than a page that is using a typical Sametime applet.

Limitations

I haven't been able to test all these platforms, but according to the documentation, Sametime Links should run with:

Windows® 95/98/NT/2000, with MSIE 5 and above (MSIE 6 with the Microsoft JVM), and Netscape 4.7x. On other browsers/platforms, the Sametime link will appear as regular HTML text. - IBM/Lotus STLinksTK.PDF

Reading this I take it to mean that it works with Microsoft Internet Explorer 5 with the SUN JVM, but I have not managed to test this. It also means there is no support for the latest Netscape browsers.

Looking at the web traffic for Instant Technologies website about 90% of users are using a compatible browser, so it tends to be a small issue to live with. Moving forward, I would hope that IBM/Lotus puts some effort into ensuring that all the Sametime components can work with the SUN JVM and the Microsoft JVM.

The IBM® Lotus® Sametime Links Toolkit 3.0 is targeted for use with the Sametime server 2.5 and above. Using the toolkit with a Sametime server 2.5 requires a server addon installation available from the Web (www.lotus.com/sametimedevolvers). The toolkit is already installed on Sametime server 3.0. - IBM/Lotus STLinksTK.PDF

Well this isn't true, currently Sametime Links only runs on Sametime 3.0 servers (I recommend upgrading straight to 3.01 and never running 3.0), although the documentation refers to being able to install Sametime Links on 2.5 servers, IBM/Lotus has never shipped this install package. You should be aware that there are number of hot fixes available for Sametime 3.01, and there is a key one that relates to Sametime Links. Without the hot fix, I found that the STMUX would be restarting as soon as approximately 3 people entered a place, apparently this is the same problem people using QuickPlace and Sametime together have been experiencing. Call Lotus Support to obtain this hot fix, otherwise you'll have to wait for Sametime 3.1 to get the fix.

Chapter 3 - Sametime Links directory overview

The first thing we should concern ourselves with when getting to know Sametime Links, is not the Javascript function calls that are available within the product, but with the contents of the Sametime Links directory that the install creates. The IBM/Lotus Sametime Links has the following:

Customizing Sametime Dialogs

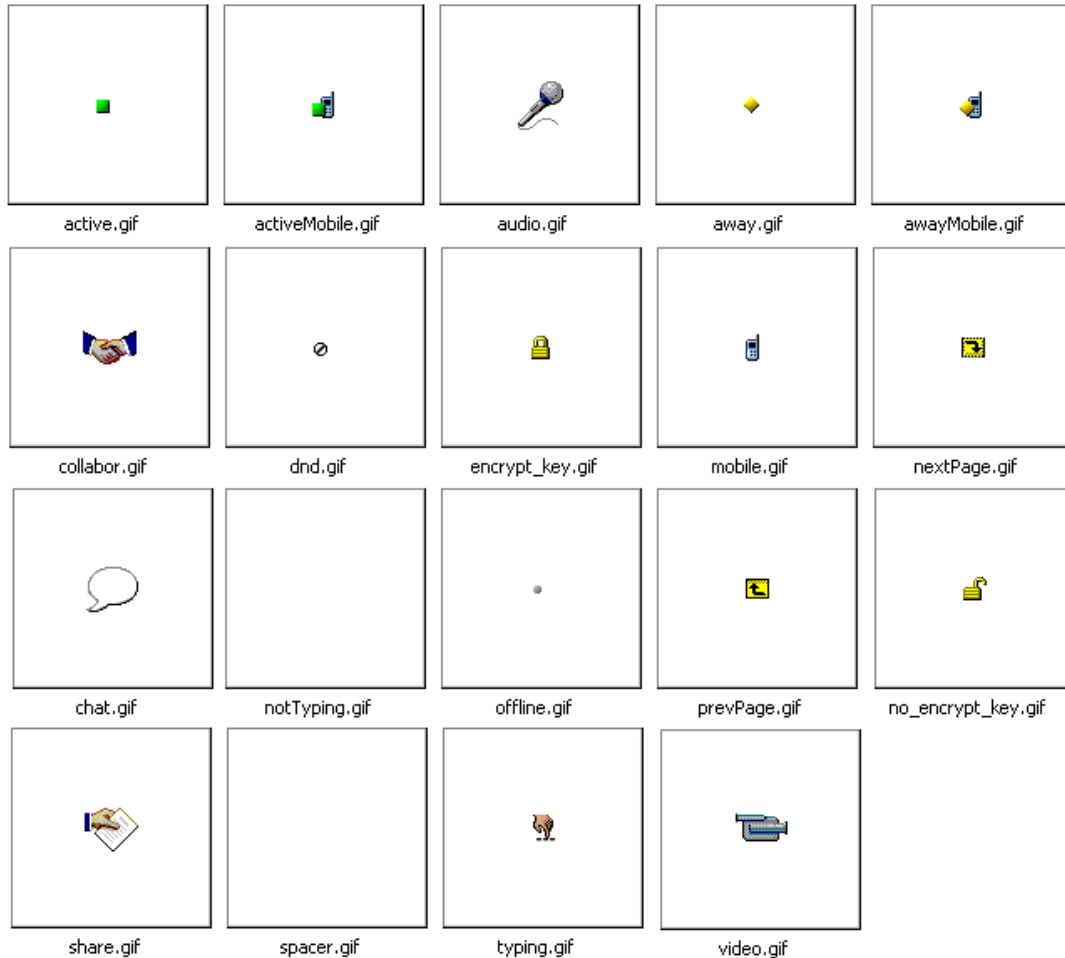
All Sametime Links dialogs are HTML pages. This allows you to easily customize the graphics; for example, you can change the background color or add your company's logo. You can also add functionality to the dialogs or disable existing functionality. For example, you can add a link to the user profile of your IM partner in the IM window. In an extranet application, you might want to prevent users from browsing the directory or inviting others to the chat. To do this, simply remove the HTML code for the "Invite Others" button from the file inputFrame.html. Similarly, to prevent users from starting Sametime meetings, remove the code for the "Add Tools" button.

Sadly that is all they really tell you about the HTML files located in the Sametime Links directory, and when you open them up there are so many nested Framesets, it can be a little daunting to figure out what goes where and which page is being used in which situation, by figuring out the contents of these directories it will make life much easier in the long run when you come to customise these pages.

Directory Contents

In the Sametime Links directory you will find a number of HTML files and directories, the directories fall into two categories

1. Language support files - "allLang", and the two letter directories, ex. "en", "de" and "es"; if you are customising the HTML files, and are supporting multiple languages you will need to modify the HTML contents of each directory. I will cover the HTML files in more detail in Chapter 4.
2. Image Directory - "img", this directory contains all the graphics used within Sametime Links



The files within the STLINKS directory are:

1. "Stlinks.js" - contains the Javascript functions used within the HTML pages to interact with the applet, and to launch the various interaction windows
2. "stlinks.cab" & "stlinks.jar" - the applet files for the MS JVM and SUN JVM respectively
3. "DebugLevel.class" - This file specifies a debug level to Sametime Links. It can't be edited and IBM/Lotus does not provide the file or specifications for it, so I wouldn't try changing it.
4. "hostInfo.js" - contains information concerning the tunnelling details for Sametime 2.5 servers, which as Sametime Links isn't available for Sametime 2.5 yet, really has no use.

5. "stlinks.css" - contains the cascading style sheet used as the default for Sametime Links pages. I won't go into the detail of this file, as it's just a simple CSS file that can be edited with notepad or a specific stylesheet editor such as TopStyle. Although I am skipping past this file, when used in combination with the contents of the "img" directory, it has great power. By making changes to the images in the "img" directory and the stlinks.css file, you can make the rather mundane looking Sametime Links, suddenly take on the look and feel of the rest of your website.

Chapter 4 - Understanding the HTML files

The HTML files that are contained within the various language sub-directories and the “allLang” directory are the key to being able to change the appearance of the Sametime Links dialogs. In this section I will attempt to outline the functions of each file, and their location, my understanding of these files came from opening and examining each one, so it is not guaranteed to be correct.

Files within the “allLang” sub-directory.

The files within the “allLang” sub-directory are common across all the languages. Changes to a file in this directory, will appear to all languages client languages.

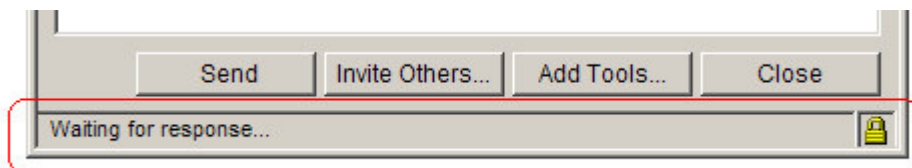
borderFrame.html

This file contains a grey background, it acts as the border in many dialogs, but also acts as a place holder to receive text through document.write commands from the Sametime Links applet, and other functions within numerous pages.

borderFrame.html is referenced in:
announcement.html
im.html
invitation.html
inviteOthers.html
nway.html
place.html
placeChat.html

statusFrame.html

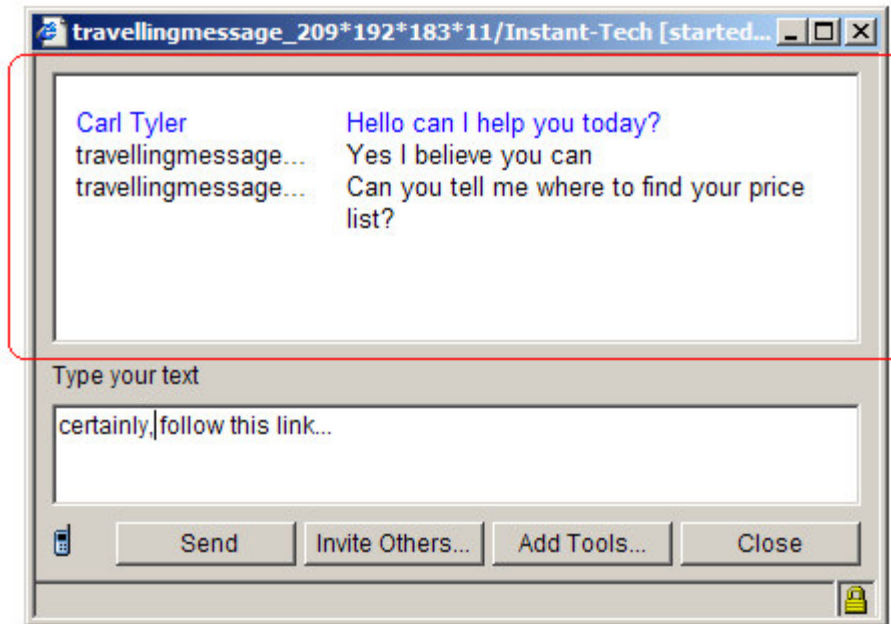
This file is used to construct the status bar at the bottom of many of the dialogs, it is also used via a parameter to display the encryption icon in the bottom right of many dialogs.



StatusFrame.html is referenced in:
announcement.html
im.html
invitation.html
nway.html
place.html
placeChat.html

transcript.html

This is used to display the chat history in various dialogs as the Instant Messaging conversation continues.



transcript.html is referenced in:

im.html

nway.html

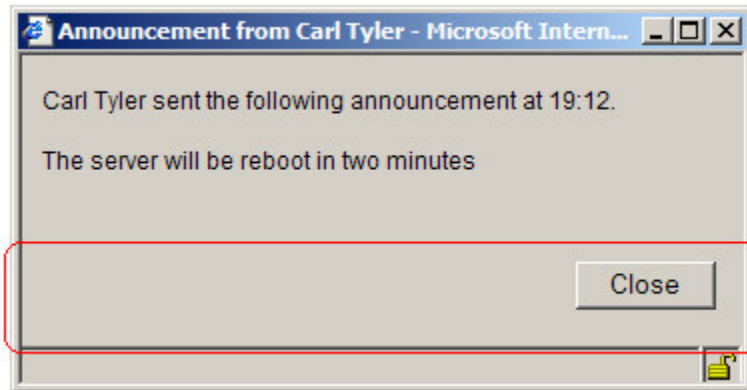
placeChat.html

Files within the individual language sub-directories.

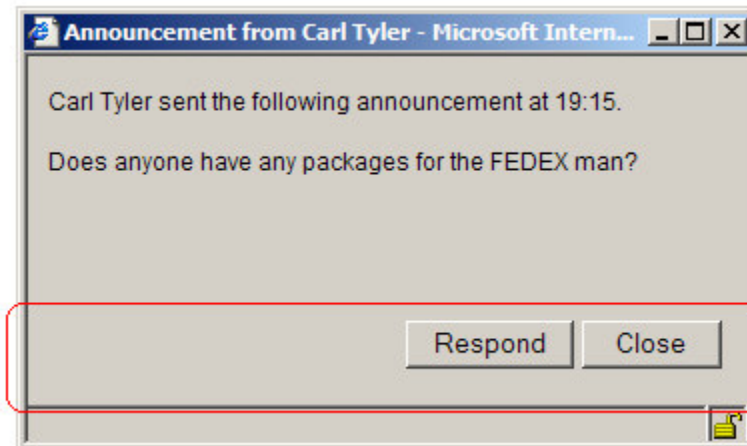
The files within the individual language sub-directories relate specifically to that country, If you are supporting multiple languages and change a file within one of these directories, you will need to change them all to be consistent across the languages. For the purpose of this documentation I will be referring to the "en" aka English directory, the filenames within the directories and their functions are the same for each country.

announceBtn.html

Contains the Respond and Close button, used with in the announceFrame.html



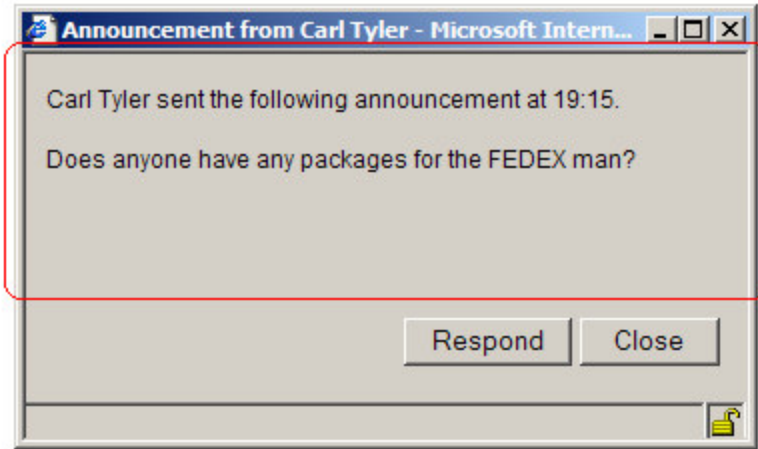
in the following screenshot, we can see the Respond button is displayed, this is displayed because the person sending the announcement had selected "Allow People to respond to me", without that option being selected the announcement would look like the screenshot above.



announceBtn.html is referenced in:
announcement.html

announceFrame.html

This is the page that passes the announcement into the borderFrame.html contained within the announcement.html dialog.

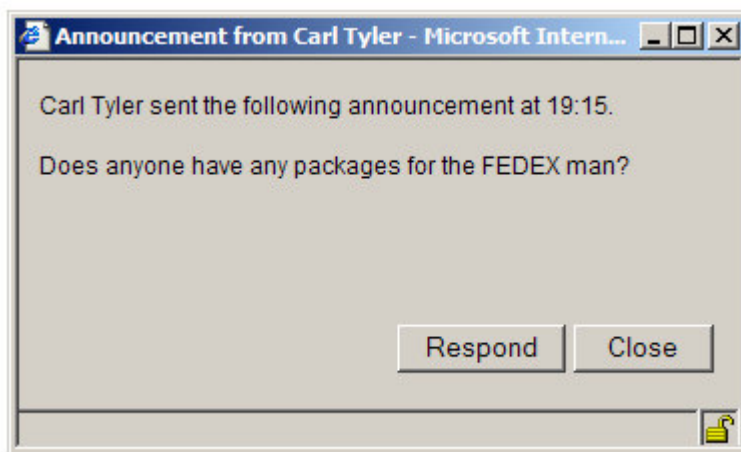


announceFrame.html is referenced in:
announcement.html

announcement.html

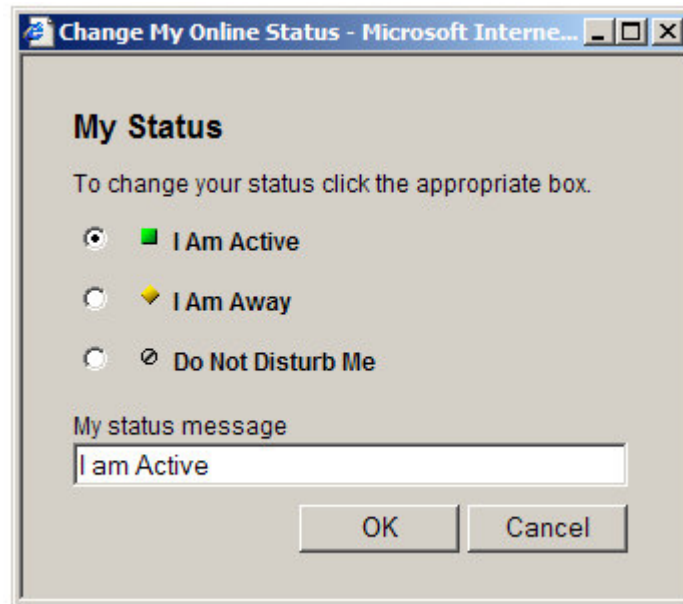
This dialog is displayed when an announcement is received. The frameset layout:

borderFrame.html (*x,*y), the contents of this frame are updated by announceFrame.html	
announceBtn.html (*x,60y)	
statusFrame.html (18x,*y)	statusFrame?secureIcon(20x,18y)



changeStatus.html

This dialog is displayed when the openStatusWindow Sametime Link is called.



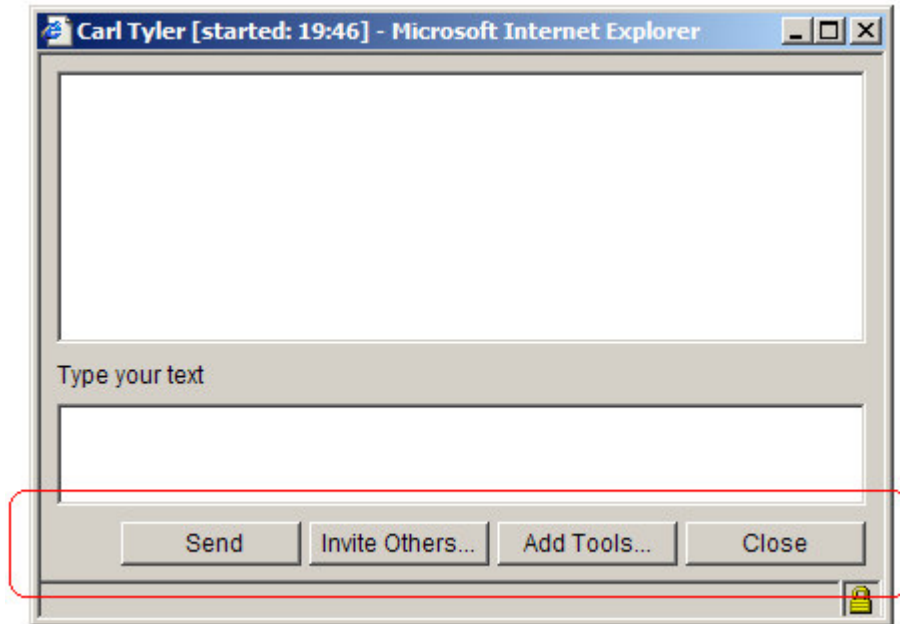
chatApplet.html

This page is never seen but is used for all chat conversations. This hidden page, contains the references to the chat functions of the Sametime Links applet.

chatApplet.html is referenced in:
chatWindow.html

chatBtn.html

This page holds the code to display the buttons "Send", "Invite Others...", "Add Tools..." and "Close"



chatBtn.html is referenced in:

im.html
nway.html
placeChat.html

chatWindow.html

This page is used to build the dialogs for:

1. Place awareness lists / "place.html"
2. Place Chat / "placeChat.html"
3. IM Chat / "im.html"
4. N-way chat / "nway.html"
5. Invitation Dialog / "invitation.html"

The frameset layout:

```
The contents of this frame are set via a parameter when launched by STLINKS.JS, the possible
contents can be:
"place.html"
"placeChat.html"
"im.html"
"nway.html"
"invitation.html" (100%x,100%)

chatApplet.html (hidden: 0x,0y)
```

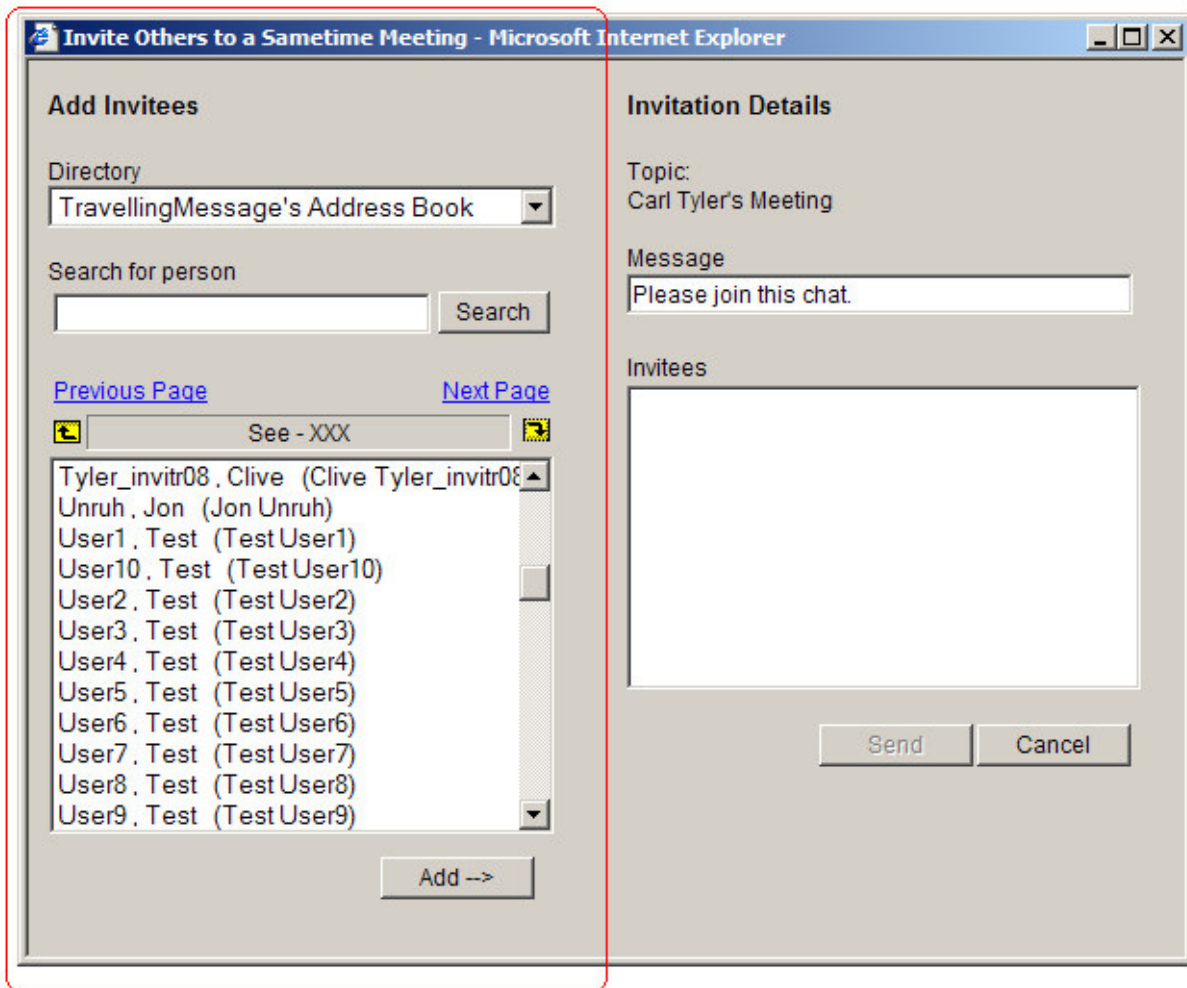
dirApplet.html

This page contains the functions for retrieving directory information when inviting users, it is not seen as it is hidden in a frameset within inviteOthers.html

dirApplet.html is referenced in:
inviteOthers.html

directory.html

This page contains the layout for the inviteOthers.html dialog. It shows the directory from the server, and allows users to be selected.

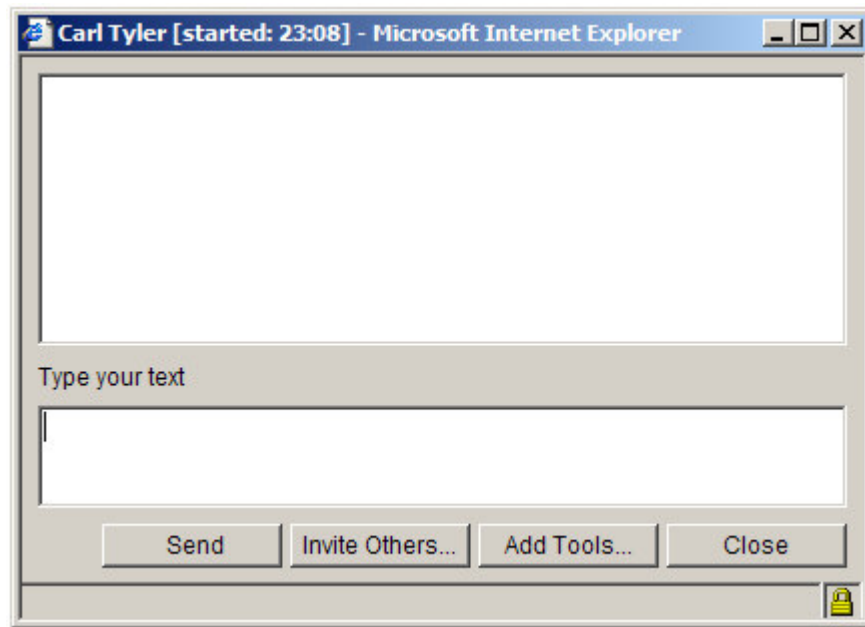


directory.html is referenced in:
inviteOthers.html

im.html

This page contains the layout for the Instant Messaging dialog when used in the chatWindow.html file.
The frameset layout:

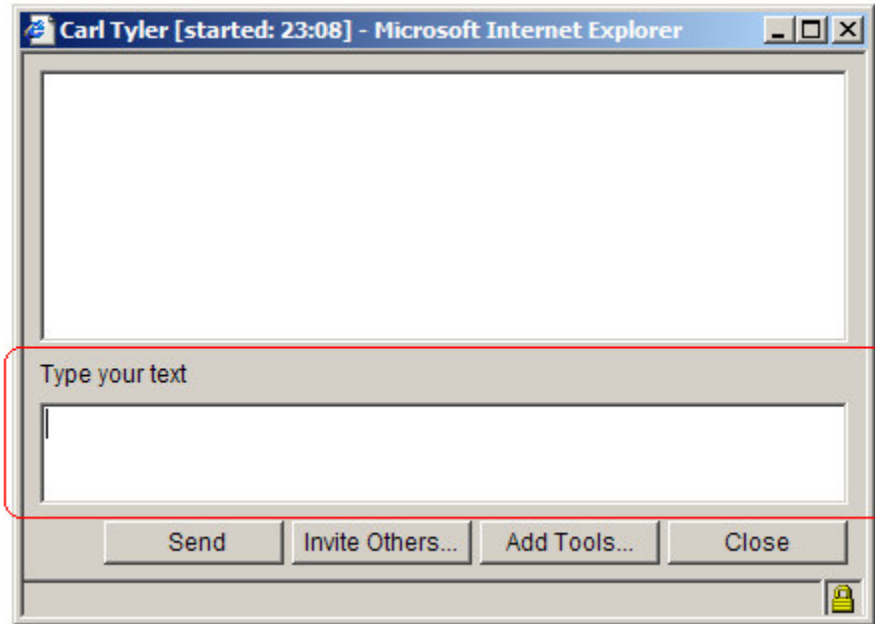
borderFrame.html (*x,8y)		
borderFrame.html(8x,*y)	transcript.html (*x,*y)	borderFrame.html(8x,*y)
	inputFrame.html (*x,82y)	
	chatBtn.html (*x,36y)	
StatusFrame.html (*x,18y)		StatusFrame?secureIcon (20x,18y)



im.html is referenced in:
chatWindow.html

inputFrame.html

This page contains the chat entry box, and checks to see if you are hitting a key etc. So that the other person can know if you are typing etc. The chat entry area is built by the function writeInputFrameApplet()



inputFrame.html is referenced in:

im.html

nway.html

placeChat.html

invitation.html

This dialog is displayed when someone is invited to a Meeting. The frameset layout:

borderFrame.html (*x,*y)	
The contents of this frame are replaced with inviteFrame.html	
StatusFrame.html (*x,18y)	StatusFrame?secureIcon (20x,18y)



invitees.html

This dialog is displayed as part of the inviteOthers.html dialog when the “Invite others...” Button is clicked.

Invite Others to a Sametime Meeting - Microsoft Internet Explorer

Add Invitees

Enter the user name, or [browse the Directory](#).

User Name

Invitation Details

Topic:
Carl Tyler's Meeting

Message

Invitees

invitees.html is referenced in :
inviteOthers.html

inviteFrame.html

This dialog is placed into the invitation.html dialog when an invite is received.



inviteFrame.html is referenced in :
invitation.html

inviteOthers.html

This dialog displays the dialog for inviting other people into a chat or meeting. The frameset layout:

resolve.html (50%x,100%y)	borderFrame.html (50%x,100%y) The contents of this frame are replaced with invitees.html
dirApplet.html (hidden:0x,0y)	

Invite Others to a Sametime Meeting - Microsoft Internet Explorer

Add Invitees

Enter the user name, or [browse the Directory](#).

User Name

Invitation Details

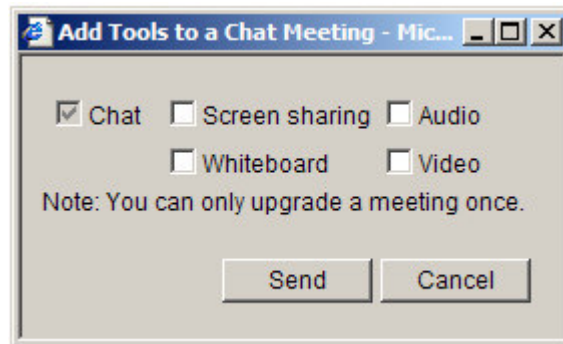
Topic:
Carl Tyler's Meeting

Message

Invitees

meetingTools.html

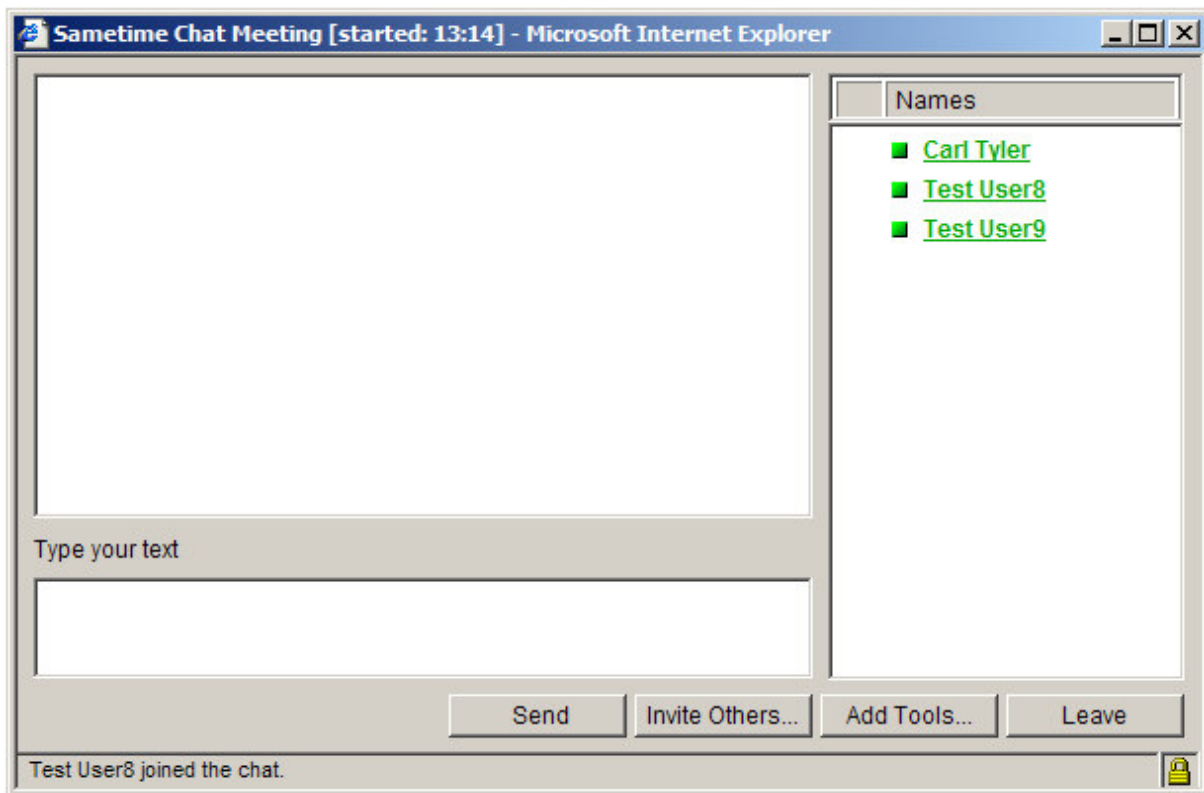
This dialog displays when adding tools to a chat.



nway.html

This page contains the layout for the nway chat dialog when used in the chatWindow.html file. The frameset layout:

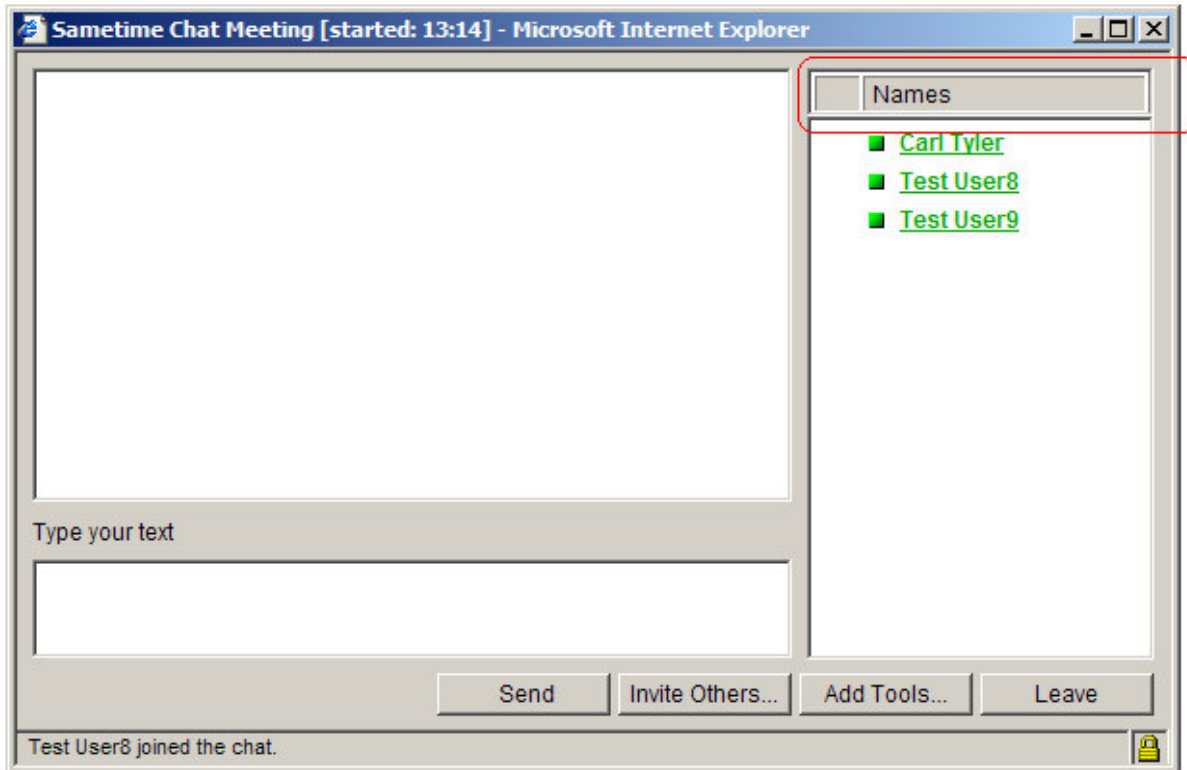
BorderFrame.html (*x,8y)				
BorderFrame.html (8x,*y)	Transcript.html (*,*)	BorderFrame.html (8x,*y)	PeopleHeader.html (180x,25y)	BorderFrame.html (8x,*y)
	InputFrame.html (180x,82y)		PeopleList.html (180x,*y)	
	ChatBtn.html (*x,36y)			
StatusFrame.html (*x,18y)				StatusFrame.html?secureIcon (20x,18y)



nway.html is referenced in:
chatWindow.html

peopleHeader.html

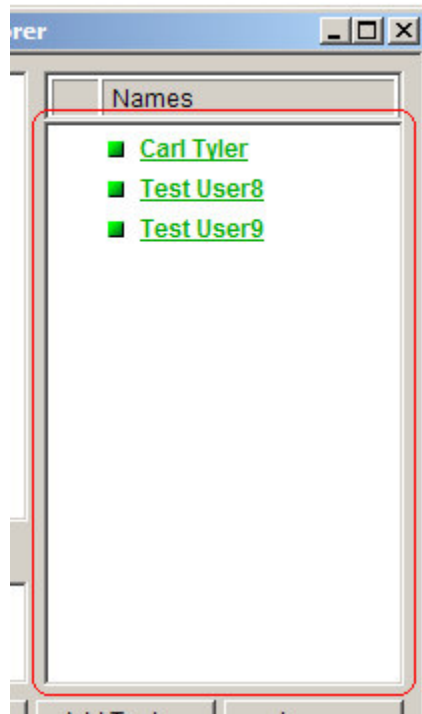
This page contains the heading for the awareness list within the nway chat and placeChat dialogs.



peopleHeader.html is referenced in:
nway.html
placeChat.html

peoplelist.html

This page contains the awareness list layout that is used in the place.html, nway.html and placeChat.html:

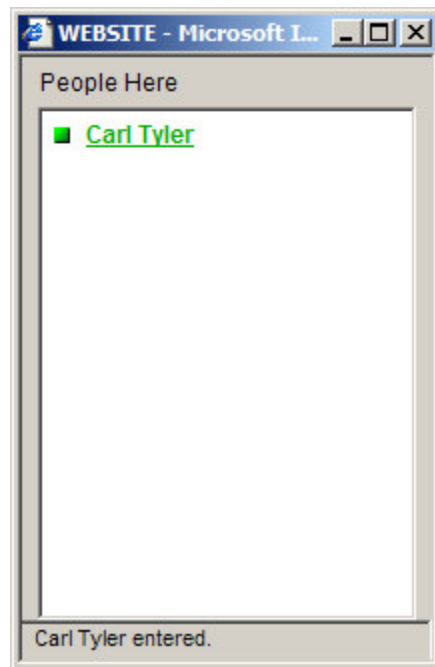


peopleList.html is referenced in:
nway.html
place.html
placeChat.html

place.html

This page contains the layout for the place awareness list, that is displayed when the function `openPlaceWin` is executed, or a Sametime Links place counter is clicked. The frameset layout:

BorderFrame.html (8x,25y)		
BorderFrame.html (8x,*y)	peopleList.html (*x,*y)	BorderFrame.html (8x,*y)
StatusFrame.html (8x,20y)		



placeChat.html

This page contains the layout for the `placeChat` dialog when used in the `chatWindow.html` file. The layout appearance is practically identical to the `nway.html` layout (see earlier) except that `placeHeader.html` is used instead of `peopleHeader.html`

`placeChat.html` is referenced in:
`chatWindow.html`

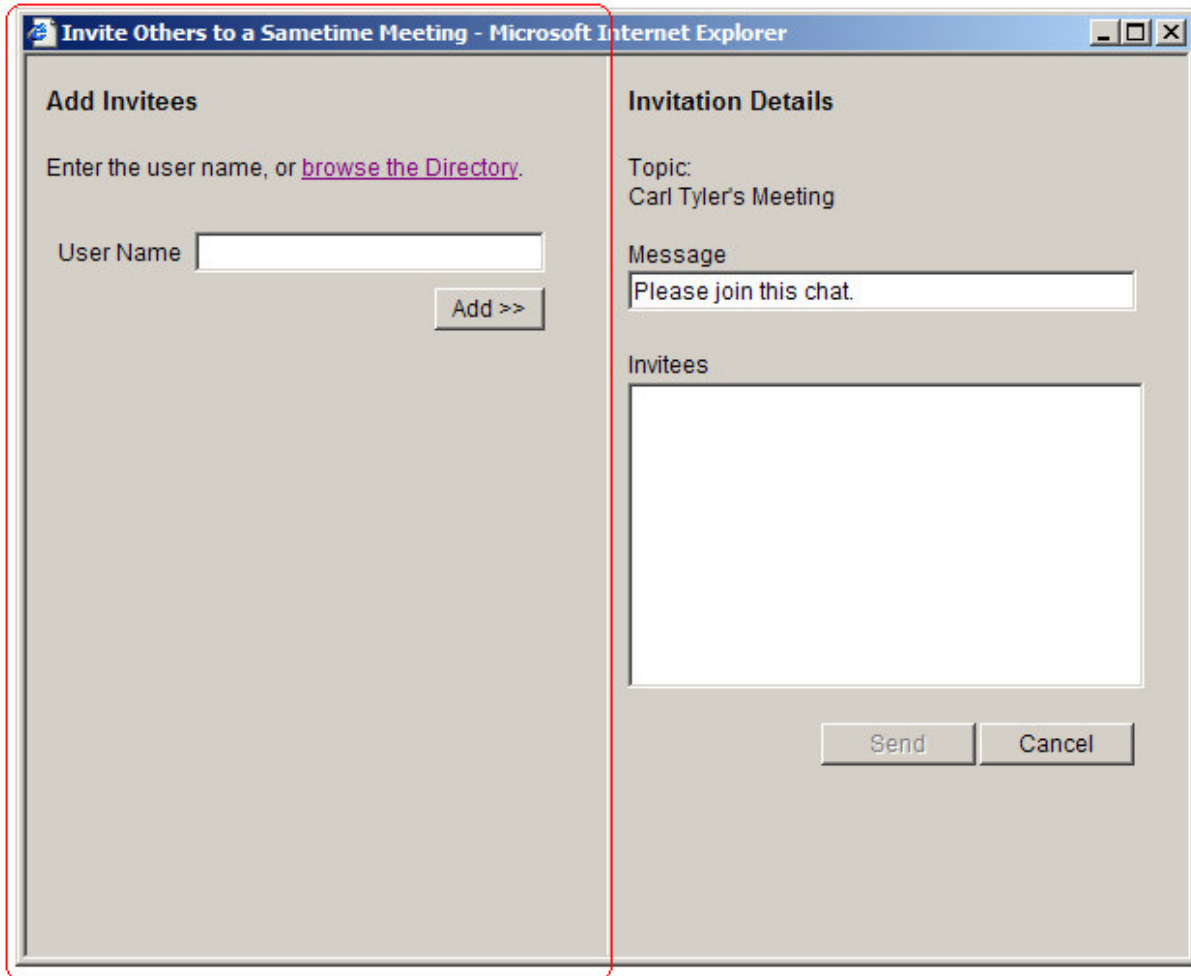
placeHeader.html

This page displays the text "People Here" as the heading above the `peopleList` in the `placeChat.html`

`placeHeader.html` is referenced in:
`placeChat.htm`

resolve.html

This page contains the layout for the Add Invitees within inviteOthers.html dialog. It allows a person to type in the user name of a person to add, or to click a button to browse the Directory, when the directory is browsed, this file is replaced with directory.html.



resolve.html is referenced in:
inviteOthers.html

res.js

Although res.js is not a HTML file, there is a copy for each language stored in the language directory, it contains a number of resource strings for use within the other pages, and also contains the default x,y sizing for many of the dialogs. By editing this file, we can change the start-up size of many dialogs, and also change some of the strings presented to the user.

Chapter 5 - Building an interactive website

So now we know where the files are, and what's in each of them we can really start to mess around. Let's go back to the original goals of our sample project:

1. Provide the customer with a customised/branded Sametime Links experience
2. Allow an agent to see who is on their web site
3. Allow the agent to interact with the customer
4. Allow the agent to see which page the customer is on
5. Let other agents know when an agent is already chatting with a customer

To make it easier to follow as we build this sample, we're going to call the internal person the "Agent" and the external visitor the "Customer", obviously in real life, the Customer could also be an internal user. Because we are going to have a separate UI for the Agent and Customer it is a good idea to have separate directories to manage them in. So we are going to take the standard STLinks directory \Lotus\Domino\Data\domino\html\sametime\stlinks and copy it and it's contents twice as:

```
C:\Lotus\Domino\Data\domino\html\sametime\agentstlinks
```

```
C:\Lotus\Domino\Data\domino\html\sametime\customerstlinks
```

Now for changes we want to make to the agent side we will edit the contents of the "agentstlinks" directory, and for customer changes we'll change the "customerstlinks" directory.

When loading Sametime Links in a web page, we can now load the page that we want, so for customer facing pages, our Sametime Links code in the header would look something like this:

```
<LINK REL=STYLESHEET HREF="<CODEBASE>/stlinks.css/" TYPE="text/css">
<SCRIPT src="<CODEBASE>/stlinks.js">
```

Where codebase is the URL of the directory where the ST Links runtime files are :

```
http://<STSERVERNAME>/sametime/customerstlinks
```

For Agent pages, the header would be the same, but the codebase would be:

```
http://<STSERVERNAME>/sametime/agentstlinks
```

Provide the customer with a customised/branded Sametime Links experience

For this sample, we are going to complete some very simple customisation to the screens that are displayed to the customer, which are:

1. Display a company logo in the top of the chat window
2. Remove the buttons to "Invite Others..." and "Add Tools..."

Display a company logo in the top of the chat window

Referring to Chapter 4, we can see that the Instant Messaging dialog layout is contained within im.html. Looking at the layout of this page, we can see the easiest way to add a logo to this page is to edit the frameset, and make the top frames height larger, and change the source file from borderFrame.html to something else, in this case we will create a new file called logoFrame.html, this new file is the same as borderFrame.html but contains our company logo, which we will place in the "img" directory. For this sample I generated a logo with dimensions of 103 by 26.

1. Place the logo file, called logo.gif into the "img" directory
2. Copy borderFrame.html in "allLang" to logoFrame.html and place in "allLang" directory

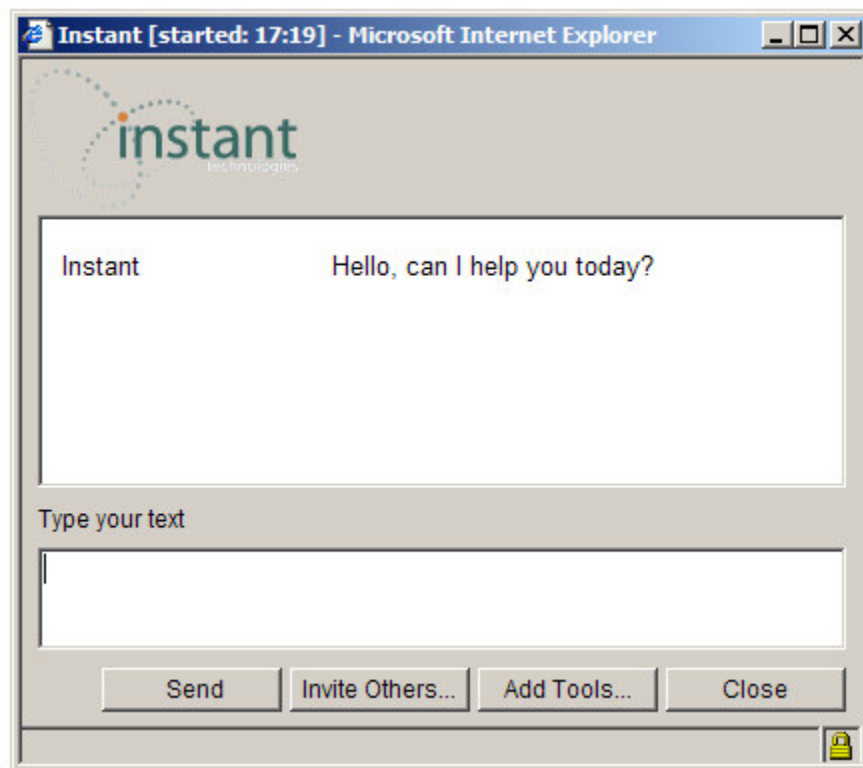
3. Edit logoFrame.html and change the lines:


```
<BODY BGCOLOR="#d4d0c8">
</BODY>
to
<BODY BGCOLOR="#d4d0c8" topmargin="0" leftmargin="0">
<p></p>
</BODY>
```
4. Open im.html in each language sub-directory, and change the top three lines from:


```
<HTML>
<frameset rows='8,* ,18' frameborder=0 border=0 framespacing=0>
<frame src=../allLang/borderFrame.html noresize scrolling=no target="_self">
to
<HTML>
<frameset rows='78,* ,18' frameborder=0 border=0 framespacing=0>
<frame src=../allLang/logoFrame.html noresize scrolling=no target="_self">
```
5. Perform these changes to the im.html in any other language directories you plan to use.
6. After making these changes, you may find that the window is too small for the transcript text, so we also want to change the default launch size for the window. Do this by opening the file res.js and changing the line


```
var WIN_IM_HEIGHT = 280;
to
var WIN_IM_HEIGHT = 350;
```

This results in the following IM chat window:



Remove the buttons to “Invite Others...” and “Add Tools...”

From Chapter 4 we can see that the buttons for the chat window are generated in the page chatBtn.html. To stop the “Invite Others...” and “Add Tools...” from appearing we will simply remark out the code that those buttons.

1. Open the file chatBtn.html in each language sub-directory and change the following lines from:

```

if (!top.isAnonymous || top.anonCanResolve)
    document.writeln('<td align=right><input type=button .... Truncated...
if (top.meetingsEnabled)
    document.writeln('<td align=right><input type=button .... Truncated...

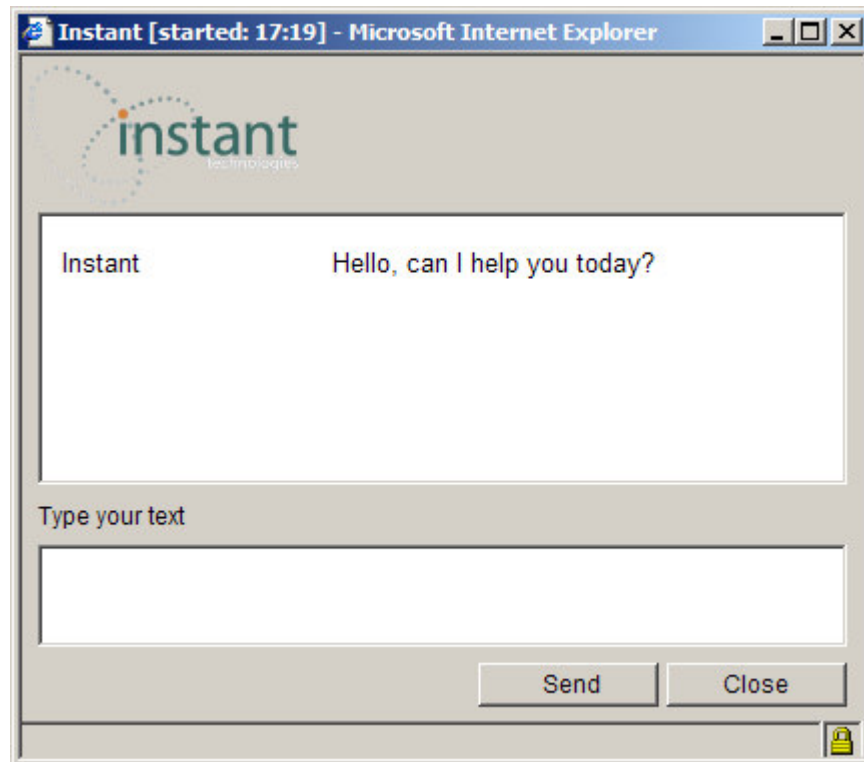
```

To

```

//if (!top.isAnonymous || top.anonCanResolve)
// document.writeln('<td align=right><input type=button .... Truncated...
//if (top.meetingsEnabled)
// document.writeln('<td align=right><input type=button .... Truncated...

```
2. This will now stop the “Invite Others...” and “Add Tools...” buttons appearing.



Allow an agent to see who is on their web site

Now that we’ve customised the experience for the customer, we need some way for the agent to see who is on line and initiate a chat with them. To keep track of who is on the website we will make use of Sametime’s ability to accept anonymous logins, and it’s ability to track places. The easiest way to think of places are as physical rooms, where when you can enter a room and see how many people are in there and who they are; one thing to keep in mind if you think this way is that with Sametime places, people do have the ability to be in more than one place at the same time, no pun intended. To keep this sample simple, we are going to have one place, that we are going to call “WEBSITE”, when a person visits a webpage, this is the place we will automatically enter them into, and this is the place the agent will watch to see who is on the website.

To complete this task there are a few steps we need to complete:

1. Modify the website pages to load the Sametime Links applet
2. Log the visitor into Sametime
3. Upon a successful login, move the customer in to the place “WEBSITE”
4. Provide a web page for the agent to login and watch visitors
5. Display on the agents page the number of people in the “WEBSITE” place

Modify the website pages to load the Sametime Links applet

We need to log the customer into Sametime via Sametime Links for each page they visit on our website and we want to track. Within the <HEAD> section of the webpage we need the following:

```
<LINK REL=STYLESHEET HREF="<CODEBASE>/stlinks.css" TYPE="text/css">
<SCRIPT src="<CODEBASE>/stlinks.js">
```

Where CODEBASE is the URL of the directory where the ST Links runtime files are :

```
http://<STSERVERNAME>/sametime/customerstlinks
```

This code provides the stylesheet and the Javascript library for the Sametime Links functions.

Adding the function setSTLinksURL is a key part of ensuring the correct HTML pages are loaded, because all the Sametime Links chat windows and other dialogs are all HTML pages, the function setSTLinksURL tells the Sametime Links where these HTML files are located on the Sametime server.

```
<SCRIPT>
setSTLinksURL("<ORIGINALCODEBASE>", "EN", "<CODEBASE>");
</SCRIPT>
```

Where ORIGINALCODEBASE is the URL of the directory where the ST Links runtime files are :

```
http://<STSERVERNAME>/sametime/stlinks
```

CODEBASE is the URL of the directory where the customized customer ST Links files are:

```
http://<STSERVERNAME>/sametime/customerstlinks
```

For complete details on stSTLinksURL function and its parameters, see page 23 of the "Sametime Links Javascript API Reference".

Log the visitor into Sametime

To log the user into Sametime we need to use the writeSTLinksApplet function with is documented in the Sametime Links developer guide. For our sample, customers are going to be logged in as anonymous users, so we need to add into the HTML of our page the following:

```
<SCRIPT>
writeSTLinksApplet ("", "", false);
</SCRIPT>
```

What this does, is logs the customer in as anonymous, their names take the form of, Userxxx/Guest where xxx is a number that increments by one for each visitor. There are ways to set the name for the visitor, but we will not cover that here, refer to the Sametime Links developer guide for more details.

Upon a successful login, move the customer in to the place "WEBSITE"

If the user is logged in successfully to Sametime, Sametime Links fires of the function STLinksLoggedIn, by waiting for this function to be called before trying to enter the place "WEBSITE" we can ensure that the Sametime Links applet has loaded successfully and avoid any Javascript errors that may occur otherwise. With the following code within the HTML page, we can log the customer into the "WEBSITE" place:

```

<SCRIPT>
function STLinksLoggedIn()
{
    STLinksEnterPlace ("WEBSITE",true);
}
</SCRIPT>

```

The first parameter to this function is the name of the place we want the customer to enter. The second parameter makes Sametime Links wait 30 seconds before taking the user out of the place. This ensures that as the customer navigates between pages, Sametime Links has time to enter the place again after the new page loads. This way, when the customer navigates between pages that enter the same virtual place, the customer will remain in the place during navigation.

Provide a web page for the agent to login and watch visitors

Now that we have the customer entering the place, we need some means for the agents to see how many people are in the "WEBSITE" place and who they are. To do that we have to perform similar steps to setting up the customers page.

First we need to log the agent into Sametime via Sametime Links on a custom page for monitoring the website. So first you need to create a standard HTML file that the agent will load. Within the <HEAD> section of this page you need the following:

```

<LINK REL=STYLESHEET HREF="<CODEBASE>/stlinks.css" TYPE="text/css">
<SCRIPT src="<CODEBASE>/stlinks.js">

```

Where codebase is the URL of the directory where the ST Links runtime files are :

http://<STSERVERNAME>/sametime/agentstlinks

As with the customer pages, we need to add the function setSTLinksURL to ensure the correct HTML pages are loaded, the function setSTLinksURL tells the Sametime Links where these HTML files are located on the Sametime server.

```

<SCRIPT>
setSTLinksURL ("<ORIGINALCODEBASE>", "EN", "<CODEBASE>");
</SCRIPT>

```

Where ORIGINALCODEBASE is the URL of the directory where the original ST Links runtime files are :

http://<STSERVERNAME>/sametime/stlinks

CODEBASE is the URL of the directory where the customised agent ST Links files are:

http://<STSERVERNAME>/sametime/agentstlinks

For complete details on stSTLinksURL function and its parameters, see page 23 of the "Sametime Links Javascript API Reference".

Next we want to have some means of getting the agent's Sametime login details. For the purpose of this sample the easiest way is to use Javascript's prompt command. So we need a few things the variables to store the agent's username and password, the prompt functions to get them from the users, and the writeSTLinksApplet to log the agent in.

This code demonstrates how to do that.

```
<script language="JavaScript1.2">
    var STUserName, STPassword;

    STUserName = prompt("Enter Sametime Login Name.", "");
    STPassword = prompt("Enter Sametime Password.", "");
    writeSTLinksApplet(STUserName, STPassword, false);
</script>
```

Display on the agents page the number of people in the “WEBSITE” place

We need a place on the page to let the agent see how many people are in the place “WEBSITE”, this is done using the Sametime Links function writePlaceCounter within the page. The following HTML on the agents page will allow you to see how many people are in the place “WEBSITE”

```
People in WEBSITE place
<script>
writePlaceCounter("WEBSITE", "WEBSITE", false);
</script>
```

The first parameter being passed to the writePlaceCounter is the name identifying the place, the second parameter is the display name for the place, this appears in the window title when the place counter is clicked on. The third parameter is if you want to keep the person in the place for 30 seconds before being seen as having left the place, as this is an agent we will set it to false. By using the function writePlaceCounter within a page, it automatically logs you into the place to get the status. So if an agent loads this page, and there are no other agents, or customers logged on to the site, the place counter will indicate “1” which indicates they are in the place. If another agent, or a customer enters the place, the counter will increase by 1 to “2”.

By clicking on the counter, the place.html window appears, show who the people are in the “WEBSITE” place. For this simple example it is easy to tell agents from customers as all the customers will have names like Userxxx/Guest.

From the place.html dialog the agent can click on a customer's name to initiate an Instant Messaging conversation with them. From that conversation it is possible for the agent to start screen sharing or send the customer URLs, or even involve other people in the conversation through the use of nway chat. If you want to really blow your customer's mind, you can start a meeting using audio/video and screen sharing to demonstrate how to do something.

Now we could stop right here, and from the customer's point of view you'd have a very powerful proactive interaction offering on your site, and if you were a single agent watching the website then you will probably find this perfectly suitable. However, in the real world there tends to be more than one agent watching the site, which leads to internal issues for the application that make it awkward in real life use such as

1. No way to know if another agent is already chatting with a customer.
2. No way to know which page the customer is on.
3. No information concerning the customer or where they came from.

Allow the agent to see which page the customer is on

There a number of ways this can be done, the simplest is to add a line to the customers HTML pages that enters them into a place with the name of the webpage so our existing HTML:

```
<SCRIPT>
function STLinksLoggedIn()
{
    STLinksEnterPlace("WEBSITE",true);
}
</SCRIPT>
```

would change to:

```
<SCRIPT>
function STLinksLoggedIn()
{
    STLinksEnterPlace("WEBSITE",true);
    STLinksEnterPlace("name of webpage here",true);
}
</SCRIPT>
```

Then on the agent's HTML page we'd build a list of writePlaceCounters on a HTML page for each webpage you have in your site, then with that single page the agent would be able to watch the number of people on the different pages, and then click on the page to see who is in there.

We also have another option which doesn't involve creating any more places, and makes use of Sametime's ability to add a status message to someone's status.

By default, visitors to your site that are logged in with Sametime Links have a status message of "userxx/Guest is active", by adding a single line of script to each webpage, we can change the customers status to have whatever information we want, such as the page the person is on, or if they are chatting with an agent already. For this example we'll start by updating the person's status whenever they load a page, so that agent can see which page they are on by simply hovering over their name in the place.html dialog.

Currently our existing customer webpage HTML contains







```
<SCRIPT>
function STLinksLoggedIn()
{
    STLinksEnterPlace("WEBSITE",true);
}
</SCRIPT>
```

By changing this to:

```
<SCRIPT>
function STLinksLoggedIn()
{
    STLinksSetMyStatus("32","On Page: " + window.location.pathname);
    STLinksEnterPlace("WEBSITE",true);
}
</SCRIPT>
```

The STLinksSetMyStatus is referred incorrectly in the Sametime Links documentation and Sametime Links release notes as STLinksChangeMyStatus, this is the same function, just documented with the wrong name, the function name you need to use is STLinksSetMyStatus. There are two parameters passed to the function STLinksSetMyStatus, the first of which is a numeric constant for representing the users status, these constants are:

● 0 - Offline

-  32 - Active
-  64 - Not using the computer
-  96 - Away
-  128 - Do not disturb (DND)
-  544 - Mobile active
-  608 - Mobile away

The second parameter is the status message, this is where you normally see messages such as “I am in a meeting.”

By changing the users status when they enter a page, and hovering over the customers name in the people list the agent can now see the whole path name of the page the customer is on, this tends to be more than what most people want to see, so with a couple of additions, we can return the last section of the path details:

```
<SCRIPT>
function STLinksLoggedIn()
{
    var currentPath = window.location.pathname;
    var currentPage = currentPath.substring(currentPath.lastIndexOf('/') + 1);
    STLinksSetMyStatus("32","On Page: " + currentPage);
    STLinksEnterPlace("WEBSITE",true);
}
</SCRIPT>
```

If you understand Javascript, you'll realise you can start to use this in combination with cookies and other variables to add other valuable information such as:

1. The time the customer first came to the site
2. The time the customer went to their current page
3. The website that referred them to your site
4. The customers hostname ex. www.microsoft.com
5. The customers IP address ex. 219.243.221.67

Let other agents know when an agent is already chatting with a customer

Currently when an agent is chatting with a customer, there is no way for other agents to know this, again making use of the status information available within Lotus Sametime, we'll make it easy for other agents to see when a customer is already chatting with an agent. As you could see earlier, there are a number of status's a Sametime Links user can be. As our customers are not going to be running Sametime Links over mobile devices, we'll make use of the 544 status, which is the Mobile Active state.


-  544 - Mobile active

To change the status when a person is in conversation with an agent, we need to grab the point where the agent starts chatting with the customer. We can intercept this through the stlinks.js file that is in the /customerstlinks directory. Within that Javascript library there is a function openImWindow. This function is called when someone sends a message to someone else. In the case of our website visitors, this can only be an agent sending them a message, so we know that when this function is called, that an agent has started a conversation with a customer. To let other agents know this, we will change the customer's state to Mobile Active, and indicate in their status message which agent they are chatting with. We do this by changing the function openImWindow in the customer stlinks.js file from:

```
function openImWindow (partnerId, partnerName, away, mobile, isSecured, winIndex, winName)
{
    partnerName=escape (partnerName);
    PartnerId=escape (partnerId);
    Open (urlPages+"/chatWindow.html?winType=IM&urlMain="+urlMain+".... Truncated
    ....
    ....
    ....
    ....
    .... Left="+winIndex*20);
}
```

To

```
function openImWindow (partnerId, partnerName, away, mobile, isSecured, winIndex, winName)
{
    partnerName=escape (partnerName);
    PartnerId=escape (partnerId);
    Open (urlPages+"/chatWindow.html?winType=IM&urlMain="+urlMain+".... Truncated
    ....
    ....
    ....
    ....
    .... Left="+winIndex*20);
    STLinksSetMyStatus ('544', 'Chatting with: ' + unescape (partnerName));
}
```

This will now display the icon  and status message "Chatting with: Agents Name" when a customer is engaged in conversation.

Chapter 6 - Conclusion

Hopefully this document has given you enough to get started, but you shouldn't stop there, using this as the basis for working with Sametime Links, I have been able to take the core Sametime Links package and offer the following capabilities to agents on our website:

1. The ability to chat with a customer, and not have the customer see the internal persons name
2. Alerts for when people enter specific pages on the website, such as support or purchase
3. The ability for a customer to request help, rather than having to wait until it is offered
4. The ability for the agent to send URLs to the customer that then automatically open in a new web browser
5. The ability to log customer visits and chats into a database

About the author

Carl Tyler is Chief Technology Officer of Instant Technologies. He has over 14 years of experience with collaboration software, including over nine years at Lotus Development and two-plus years at IBM Development UK. Whilst at Lotus held a number of positions such as World-wide Knowledge Management Manager, Lotus Notes R5 Marketing Manager, Internet SmartSuite Marketing Manager, Senior Systems Engineer for the UK Northern Region. After leaving Lotus Development, Carl worked for UK-based Virtual Village as VP of Development in their US office, Virtual Village developed solutions for the recruitment industry and in 2000 was acquired for 10m Euro by JobPilot, whilst at Virtual Village Carl managed worked with a small team of developers to build a hosted Sametime support offering, built from the ground up the application offered customer queues, offline query handling, automatic archiving and the ability to build a knowledge base on the fly from customer questions. Prior to starting Instant Technologies with Peyton McManus, Carl worked for Principle Software, building the first commercial chat archiving tool for Lotus Sametime.

About Instant Technologies

Instant Technologies develops software solutions that embed the speed, convenience, and utility of secure, enterprise instant messaging (eIM) within leading business software applications. With award winning solutions that target the eIM-to-application integration opportunity, Instant Technologies, allows you to receive the benefits of enhanced efficiency and productivity through the intelligent integration of eIM into the communications workflow of your enterprise.

In addition to its product offerings, for organisations needing advice or assistance on extending their instant messaging or real time communication infrastructure, Instant Technologies can help ensure you maximise your instant messaging investment with our full compliment of consulting services.

Instant Technologies
[Http://www.instant-tech.com](http://www.instant-tech.com)
Tel: 1-978-922-4029
Fax: 1-978-922-4652
e-mail: info@instant-tech.com