Adding a popup menu to your Sametime links

Level: Intermediate
Works with: Sametime
Updated: 01-Jul-2003

by
Haim
Schneider

The Sametime Links Toolkit is an easy and cool way to enrich Web applications with online awareness and instant collaboration. By adding just a few lines of HTML code to your Web application, existing names are turned into live Sametime links. By now, you may have already enabled your application with Sametime links. But now that the names are live, maybe you want your live names to do more. In addition to sending a message, you may want your application to offer more options. Perhaps you want to send email to an offline person. Or you want to add the name to your contact list, view the person's profile, or maybe invite someone to an audio meeting. The good news is that the Sametime Links Toolkit allows you to do all that quite easily. This article shows you how to override the click behavior of Sametime links and how to display a popup menu when a link is clicked. Using dynamic HTML and the Sametime Links JavaScript API, we create a popup menu that provides multiple options related to the selected person.

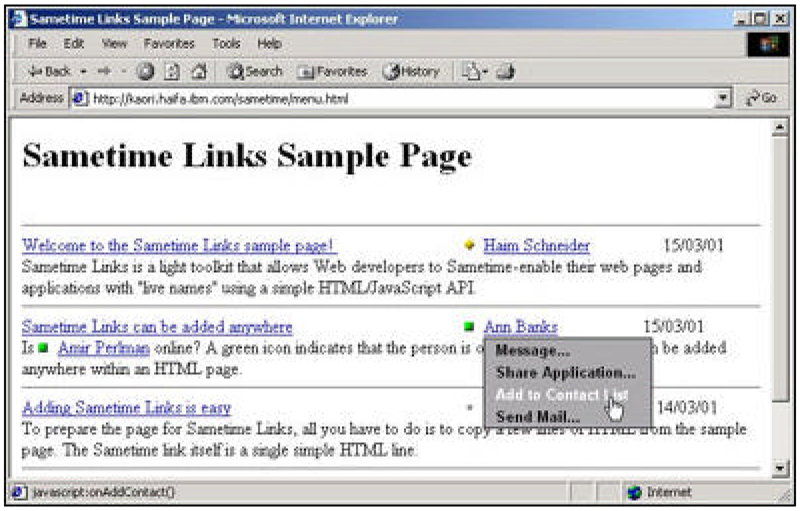
This article assumes that you have a basic understanding of dynamic HTML and JavaScript and that you are familiar with the Sametime Links Toolkit.

The sample popup menu

The sample popup menu that we are going to add has the following four menu items:

- *Message*
  Sends an instant message to the selected person.
- *Share Application*
  Invites the selected person to an application-sharing meeting.
- *Send Mail*
  Creates a new email message for the selected person with the user's default email client.
- *Add to Contact List*
  Adds the selected person to the user's Sametime Connect's contact list.

The following screen shows the sample popup menu.


Sametime Links Sample Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://kaori.haifa.ibm.com/sametime/menu.html
Go
Message...
Share Application...
Add to Contact List
Send Mail...
Internet

Sametime Links Sample Page

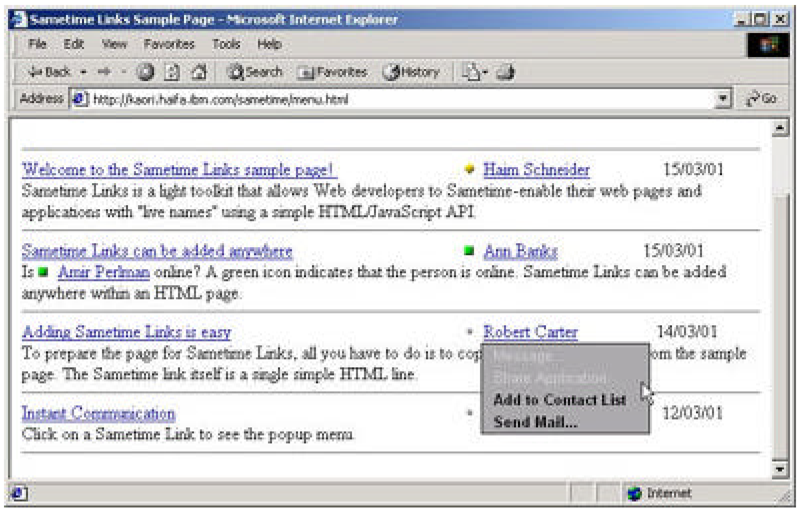
Welcome to the Sametime Links sample page! Haim Schneider 15/03/01
Sametime Links is a light toolkit that allows Web developers to Sametime-enable their web pages and applications with "live names" using a simple HTML/JavaScript API.

Sametime Links can be added anywhere Ann Banks 15/03/01
Is Amir Perlman online? A green icon indicates that the person is o... be added anywhere within an HTML page.

Adding Sametime Links is easy 14/03/01
To prepare the page for Sametime Links, all you have to do is to copy a few lines of HTML from the sample page. The Sametime link itself is a single simple HTML line.

javascript:onAddContact()

Sametime links are normally displayed as simple HTML text when the person is offline. They are not clickable links because you cannot send an instant message to an offline person. But after introducing the popup menu, this will change. We want the menu enabled also for offline names because some of the options are applicable to offline people. You can send email to an offline person or add that person to your contact list. Therefore, we will change the default behavior of the links and make offline names display as HTML links. When an offline name is clicked, the menu items that are only applicable to online people are disabled, as you can see in the following screen.


Sametime Links Sample Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://kaori.haifa.ibm.com/sametime/menu.html
Go
Message
Share Application
Add to Contact List
Send Mail...
Internet

Welcome to the Sametime Links sample page! Haim Schneider 15/03/01
Sametime Links is a light toolkit that allows Web developers to Sametime-enable their web pages and applications with "live names" using a simple HTML/JavaScript API.

Sametime Links can be added anywhere Ann Banks 15/03/01
Is Amir Perlman online? A green icon indicates that the person is online. Sametime Links can be added anywhere within an HTML page.

Adding Sametime Links is easy Robert Carter 14/03/01
To prepare the page for Sametime Links, all you have to do is to cop... om the sample page. The Sametime link itself is a single simple HTML line.

Instant Communication 12/03/01
Click on a Sametime Link to see the popup menu.

The menu is presented as a sample only and is easily customizable. This article shows you how to implement the menu framework and how to add menu items. You can use the menu as is or add or replace items with others that you want to present in your application. You can add application-specific options, for example, showing a list of documents authored by the selected person or displaying a user profile.

Starting with a Sametime links-enabled page

In this article, we start with a Web page that is already enabled with Sametime links and add the popup menu to it. If your application is not yet enabled with Sametime links, see the instructions in the **Sametime Links Toolkit Developer's Guide**. It's really easy; basically, you add several simple lines of HTML/JavaScript code to the Web page and provide the user's login name and password or single sign-on token. You can also start by adding the popup menu to one of the sample pages provided with the toolkit.

To access the Developer's Guide and the samples, go to the Sametime Links Toolkit home page. You can go to the toolkit home page by selecting the SDK link in the Sametime server home page. In the toolkits page, select the Sametime Links Toolkit link. If there's no link, then you first have to install the Sametime Toolkits package on the Sametime server. You can also find the Developer's Guide and the toolkit download on the **Toolkits and Drivers page**.

Note that the Sametime Links run-time files are pre-installed on the Sametime 3.0 server, so you only need to install the toolkit for the samples and the Developer's Guide. If you are using a Sametime 2.5 server, you have to install Sametime Links for Sametime 2.5.

Making offline names clickable links
By default, offline Sametime links are displayed as normal HTML text. To provide a menu for offline Sametime links, we need to make offline names clickable. We do this by adding the offlineLink option to the writeSametimeLink call:

```
writeSametimeLink("Haim Schneider/Haifa/IBM", "Haim Schneider", true, "offlineLink:yes")
```

This option should be used only if you want to override the click behavior (which we are going to do in the next step). Otherwise, when the link is clicked, Sametime links would start an instant message with the offline user, which would, of course, fail.

With the offlineLink option, offline names are displayed as normal HTML links—blue underlined text. There are now two styles of links—bold green links for online people and normal blue links for offline people. The mixed style may be confusing, so we are going to change the style of online links and make it the same style used for offline links. Although the green bold style is normally used to highlight online names, the status icon next to the name is by itself a good indication of the online status.

The styles for online and offline links are defined in the cascading style sheets file, stlinks.css. You can modify the online style in this file, and this will change the styles on every Web page that links to this style sheet. Alternatively, you can override the style in your page by adding the following style definition:

```
<STYLE>
a.online:link {color:blue; font-weight:normal}
a.online:visited {color:blue; font-weight:normal}
a.online:hover {color:blue; font-weight:normal}
a.offline {color:blue; font-weight:normal}
</STYLE>
```

Put this style definition after the <LINK> tag that links to stlinks.css so that it overrides the online style definition in that file.

Handling the click event
We use the event STLinkClicked to override the default click behavior. To handle one of the event types of the Sametime Links Toolkit, you add a function with the name of the event to your page. The function should have the parameters specified by the event type. To handle the click event, we add an implementation of STLinkClicked. This function is called whenever a Sametime link is clicked. Note that unlike all other events in the toolkit, STLinkClicked overrides the default behavior. Instead of starting an instant message, which is the default behavior, our function is called. For all other events, custom event handlers are called after the default handler is called.

The STLinkClicked event parameters include the following:

Parameter	Description
userName	The unique user name as provided in the writeSametimeLink call
displayName	The display name of the user
status	A numeric constant indicating the current status of the user (See the list of status values in the Developer's Guide.)

evt	The JavaScript Event object that provides information on the click event, such as the mouse position

We use a global variable called selectedName that stores the name of the user who was clicked. This variable is used by the function that draws the menu and the functions that implement the menu items actions.

```
var selectedName = "";

function STLinkClicked(userName, displayName, status, evt)
{
        selectedName = userName;
        showMenu(userName, displayName, status, evt.clientX, evt.clientY);
        evt.cancelBubble = true;
}
```

Our implementation of the STLinkClicked event simply sets the selectedName variable and calls showMenu to display the menu. We also have to set the event object's cancelBubble property to true in order to cancel bubbling of the event beyond the link itself.

Displaying the menu

Before adding the menu HTML code, we add style definitions for the menu box, a menu item, and a disabled menu item. Add the styles at the head section of the page.

```
<STYLE>
.stMenu {background-color:darkgray; position:absolute; visibility:hidden; border-Width:1px; border-Style:solid;
border-Color:black}
a.menuItem {color: black; font: 10pt Arial; font-weight:bold; text-decoration:none;}
a.menuItem:hover {color: white}
.menuDisabled {color: lightgrey; font: 10pt Arial; font-weight:normal; text-decoration:none;}
</STYLE>
```

We made the menu background dark gray with black bold text for the menu items. The text of a highlighted menu is displayed in white, and disabled menu items are displayed as light gray plain text. The colors, fonts, and other style attributes can be easily modified, so if you prefer your menus green on a purple background with a yellow border, just change the style definitions. Note also that we use the text-decoration: none attribute value to remove the underline from the links. HTML links are underlined by default, and we have to remove the underline to make the text look more like a menu item.

Now that we have the styles defined, we can add the menu. The menu is a <DIV> tag that contains a table, and each menu item is a table cell. You can place the menu <DIV> tag anywhere in your page. It's initially hidden because it uses the stMenu style, which has a visibility: hidden attribute.

```
<DIV id=menu class=stMenu>
        <TABLE border=0 cellPadding=2 cellSpacing=0 width=150>
                <TR><TD id="menu_message"></TD></TR>
                <TR><TD id="menu_meeting"></TD></TR>
                <TR><TD id="menu_addContact"></TD></TR>
                <TR><TD id="menu_mail"></TD></TR>
        </TABLE>
</DIV>
```

The menu is initially empty. There are four table cells for the four menu items, but these cells are empty. The cells are dynamically populated when the menu is opened. This allows us to enable or disable the menu items according to the online status and to create the Send Mail menu item with the email address of the selected person.

The setMenuItem function sets the content of a single menu item.

```
function setMenuItem(itemId, text, action, isEnabled)
{
	var menuItem = document.getElementById(itemId);
	//add spaces to the left of the text
	text = "&nbsp " + text;
	if (isEnabled)
		menuItem.innerHTML = "<A class=menuItem + href='" + action + "'>" + text + "</A>";
	else menuItem.innerHTML = "<SPAN class=menuDisabled>" + text + "</SPAN>";
}
```

The ID of the table cell (the <TD> tag) is used to access the item. SetMenuItem populates the cell with the menu item content. An enabled item is displayed as an HTML link with the action specified in the action parameter. The action is either a JavaScript URL or a mailto: link. A disabled item is displayed as plain text, using the menuDisabled style, which we defined above.

Now let's look at the showMenu function.

```
function showMenu(userName, displayName, status, x, y )
{
	var isOnline = (status != 0);

	setMenuItem("menu_message", "Message...", "javascript:onMessage()", isOnline);
	setMenuItem("menu_meeting", "Share Application...", "javascript:onMeeting()", isOnline);
	setMenuItem("menu_addContact", "Add to Contact List", "javascript:onAddContact()", isLoggedIn);
	setMenuItem("menu_mail", "Send Mail...", "mailto:" + getEmailAddress(userName), true);

	var theMenu = document.getElementById("menu");
	theMenu.style.left=x - 10 + document.body.scrollLeft;
	theMenu.style.top=y + 10 + document.body.scrollTop;

	theMenu.style.visibility='visible';
}
```

Before displaying the menu, showMenu sets the content of the four menu items. It positions the menu at the point on the screen where the mouse was clicked, and finally makes the menu visible.

The first two items, Message and Share Application, are disabled if the status parameter is zero, which means that the selected person is offline. The Add to Contact List item is disabled if the page user is not logged in. The user has to be logged in with the Sametime Links applet in order to add a person to the contact list. In showMenu, we use the flag isLoggedIn to test whether or not the user is currently logged in. This flag is not provided by the API. Because there is no direct API call for testing if the user is logged in, we use the STLinksLoggedIn and STLinksLoggedOut API events to set the isLoggedIn flag.

```
var isLoggedIn = false;

function STLinksLoggedIn(myUserId, myUserName)
{
	isLoggedIn = true;
}

function STLinksLoggedOut(reason)
{
	isLoggedIn = false;
}
```

Menu options implementation

Next, we implement the functions that are used as the actions for our sample menu items, onMessage, onMeeting, and onAddContact. As you can see in the following, these functions are very easy to implement with the Sametime Links Toolkit.

```
function onMessage()
{
	STLinksCreateIM(selectedName);
}

function onMeeting()
{
	STLinksCreateMeeting(selectedName, "chat;share;whiteboard", "Meeting", "Please join the meeting");
}

function onAddContact()
{
	STLinksAddToContactList(selectedName, "Work" );
}
```

The function onMeeting creates an application-sharing meeting with the selected person. The parameters passed to STLinksCreateMeeting are the names of the invitees, the list of meeting tools (we selected chat, application sharing, and whiteboard), the meeting topic, and the invitation message.

The function onAddContact adds the selected person to the user's contact list. The name is added to the Work group. If the user does not have a personal group named Work, the group is automatically created. For simplicity sake, we used a hard coded group name, but if you want, you can let the user choose which group the user is added to. The Sametime Links Toolkit allows you to get the list of the user's personal groups. To get the list of groups, you call STLinksGetPrivateGroups. The result is returned in the STLinksPrivateGroupsReceived event. After you have the list of personal groups, you can change onAddContact so that it presents a dialog allowing the user to choose a group from the list of existing ones or enter a new personal group name.

Note: Getting the list of personal groups and adding a user to a personal group are the only things you can do with the contact list using the Sametime Links Toolkits. You won't find in this API other functions for retrieving the contact list or manipulating it, for example, adding public groups or removing users. These APIs were not added in order to keep the Sametime Links applet as small as possible. The Sametime Links Toolkit provides the main functionality needed for a client-side application. If you want to retrieve the contact list or manipulate it in your Web application, consider doing it on the server-side using the Sametime Community Server Toolkit. In a client application, you can use the Sametime Java Toolkit to access the contact list.

Getting the email address

The Send Mail item is actually an HTML mailto link with the email address of the selected person. Where does the address come from? Well, there's no magic way to get it. The email address has to be placed on the page when the page is generated. We assume that the application that generates the page puts a mapping of names to email addresses in the page. This mapping should include all the names that have Sametime links in the page. We implement the mapping with a JavaScript object. For example, if Robert Carter and Ann Banks are Sametime links in the page, the application that generates the page adds JavaScript code that maps these names to their email addresses.

```
<SCRIPT>
	var emailAddress = new Object();
	emailAddress["Robert Carter"] = "rcarter@acme.com";
	emailAddress["Ann Banks"] = "abanks@acme.com";
</SCRIPT>
```

The names used in this mapping should be the same as the strings that are passed in the first parameter of the writeSametimeLink call. The function getEmailAddress is used by showMenu to get the email address of the selected person.

```
function getEmailAddress(name)
{
	return emailAddress[name];
}
```

Closing the menu
We're almost done—the only thing left is to make sure that the menu is closed. Like any other menu, our menu needs to be closed when a menu item is selected or the mouse is clicked outside the menu area. We add a hideMenu function that hides the <DIV> tag when the menu needs to be closed.

```
function hideMenu()
{
	document.getElementById("menu").style.visibility='hidden';
}
```

How does hideMenu get called? The simplest way is to call it in the onClick attribute of the page's <BODY> tag.

```
<BODY onClick=hideMenu()>
```

Alternatively, if you don't want to change the <BODY> tag, you can set the onClick action with JavaScript. Add the following code anywhere in the page:

```
<SCRIPT>
	document.onclick = hideMenu;
</SCRIPT>
```

How to use the sample page
The sample page available with this article from the **Sandbox** is based on one of the sample pages provided with the Sametime Links Toolkit. The original sample demonstrates how Sametime links can be embedded in a message board page. For this article, we enhanced the page with our sample menu. Don't try to run the sample like that. It contains placeholders for the server address and people names that you have to complete before you can run it.

First, in the head section of the page, replace the three appearances of <your Sametime server> with the host name of the Sametime server. For example, if the host address of the Sametime server is sametime.acme.com, the head section of the page should contain the following:

```
<LINK REL=STYLESHEET HREF=http://sametime.acme.com/sametime/stlinks/stlinks.css TYPE="text/css">
<SCRIPT src="http://sametime.acme.com/sametime/stlinks/stlinks.js"></SCRIPT>
<SCRIPT>
	setSTLinksURL("http://sametime.acme.com/sametime/stlinks");
</SCRIPT>
```

In order for the links to show online users, change the Sametime link parameters Name1 through Name5 to the names of people who are registered in your directory. For example, replace:

```
writeSametimeLink("Name1", "Name1", true, "offlineLink:yes")
```

with

```
writeSametimeLink("Ann Banks", "Ann Banks", true, "offlineLink:yes")
```

To ensure the names are unique, you can use the canonical name format. For example:

```
writeSametimeLink("Ann Banks/HR/Acme", "Ann Banks", true, "offlineLink:yes")
```

Fill in the email address mapping. There are five mapping assignments for Name1 through Name5. Replace the names with the names that you used in the calls to writeSametimeLink. Replace the email addresses with the corresponding email addresses. For example, replace:

```
emailAddress["Name1"] = "email1";
```

with

```
emailAddress["Ann Banks"] = "bann@acme.com";
```

Then replace the login name and password in the writeSTLinksApplet call:

```
writeSTLinksApplet("<login name>", "<password>", false)
```

with the login name and password of a registered user. Note that although this sample uses password authentication, most Web applications enabled with Sametime links use token authentication to achieve single sign-on and better security. See the Developer's Guide for instructions on how to use token authentication.

Put the page on a Web server and open it in a browser. You must use the URL, not the local file system address, to access the page. Sametime links are not fully functional when you access an enabled page as a local file. When an enabled page is accessed as a local file, the Sametime links are active, but a JavaScript error is displayed when you try to open a message window.

Conclusion

Adding a popup menu to Sametime links allows you to add more functionality to live names in your application. This article walked you through the code you need to add to your page to implement the menu. In doing so, we covered several topics related to Sametime links customization and the HTML/JavaScript API. Here's a list of the API topics that we covered:

- Displaying offline names as HTML links
- Changing the style used to display online names
- Overriding the click behavior
- Handling API events
- Using the API to add a user to the contact list, to create an instant message, and to create a meeting

The popup menu sample is a good example of the simplicity of the API. The sample goes beyond basic Sametime Links-enabling into what the Developer's Guide calls the advanced API, but as you can see, the advanced API is still easy to use. A single line of JavaScript code is all you need to do things like adding a person to the contact list, creating an instant message, or creating an application-sharing meeting.

ABOUT THE AUTHOR
Haim Schneider is the architect of the Sametime Links Toolkit. He has been with the Sametime development team in Rehovot, Israel since the beginning of Sametime, after his former company, Ubique, was acquired by IBM in 1998. Haim has worked on various parts of the Sametime community services, focusing mainly on the toolkits. He has an MS in Computer Science from the Weizmann Institute of science and a BS in Physics and Mathematics from the Hebrew University of Jerusalem.