

Deploying SALT Telephony Call Control on an e-Business Site

Glen Shires
Director, Media Server Architecture
Intel Communications Group
Intel Corporation

Jim Trethewey
Staff Software Engineer
Intel Labs
Intel Corporation

Table of Contents

(Click on page number to jump to sections)

DEPLOYING SALT TELEPHONY CALL CONTROL ON AN E-BUSINESS SITE	3
OVERVIEW	3
THE SCENARIO.....	3
EASY TO ADD FEATURES	4
CABLE, DSL, WIRELESS	5
PHONE NUMBER FORMAT	6
DROPPED CALLS	6
YOU'VE BEEN FRAMED.....	6
SUMMARY.....	7
MORE INFO.....	7
AUTHOR BIOS	7

DISCLAIMER: THE MATERIALS ARE PROVIDED "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT OF INTELLECTUAL PROPERTY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT SHALL INTEL OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, LOSS OF INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE MATERIALS, EVEN IF INTEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISDICTIONS PROHIBIT THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU. INTEL FURTHER DOES NOT WARRANT THE ACCURACY OR COMPLETENESS OF THE INFORMATION, TEXT, GRAPHICS, LINKS OR OTHER ITEMS CONTAINED WITHIN THESE MATERIALS. INTEL MAY MAKE CHANGES TO THESE MATERIALS, OR TO THE PRODUCTS DESCRIBED THEREIN, AT ANY TIME WITHOUT NOTICE. INTEL MAKES NO COMMITMENT TO UPDATE THE MATERIALS.

Note: Intel does not control the content on other company's Web sites or endorse other companies supplying products or services. Any links that take you off of Intel's Web site are provided for your convenience.

Deploying SALT Telephony Call Control on an e-Business Site

Glen Shires
Director, Media Server Architecture
Intel Communications Group
Intel Corporation

Jim Trethewey
Staff Software Engineer
Intel Labs
Intel Corporation

Overview

Speak to a Web page? Telephony call control for e-Commerce? The technology has arrived. Developers can now add these features to both new and existing e-applications. SALT (speech application language tags) lets users talk (replacing or supplementing the keyboard, mouse, or stylus) to access information online, order products, and so on. The telephony call-control features also let users make or receive calls through their computer or participate in phone conferences directly from a Web page in their browser.

Call-control features are available in a JavaScript* “include” library file named “saltcc.js”. “saltcc.js” is royalty-free, modifiable source code available under license from Intel. To embed telephony features on your Web site, simply place this file on your Web server, and add the mark-up tags to your Web pages so that the user’s PC can make the call. Along with the use of a call-control frame and good use of the target= attribute, SALT gives developers robust telephony features for online customers.

The Scenario

Consider the following scenario. Your company has developed and deployed, at considerable expense, an e-Commerce Web site for selling products online. Unfortunately, the order rate is not meeting expectations. When you look at how many shoppers actually purchase products, you find that, although the site has a large number of visitors, only a small percentage ever follow through on the buying process.

Now that the new SALT HTML tags are available, there is an easy way to help customers resolve problems that are keeping them from completing a purchase. Using SALT telephony call-control features, simply add a “click to dial” capability to your site (see Figure 1). With this feature, users click a single button, which temporarily turns their PC into a telephone. When the button is clicked, the user’s PC places a telephone call (using the Internet) to your company’s support line (or some other specified phone line). The telephony features allow the phone call to occur simultaneously with the Web application. This lets support personnel answer questions and help resolve orders even as users continue to view and search the pages on that site.



Figure 1. Example graphic for SALT telephony call-control feature.

Easy to Add Features

Adding telephony call control to an e-Business site is simple:

1. Include a reference to the SALT call-control library within your Web page.
2. Create the “click to dial” buttons and images, and add them to the Web page.
3. Add JavaScript functions that respond to the button click, using the library that will place the actual telephone calls.
4. Install the SALT call-control “include” library on your Web server.

SALT call-control features are embodied in a JavaScript “include” library with the file name “saltcc.js”. This is the library that must be installed on your Web server. Section 4 of Figure 2 shows that the library is installed at the top level of the business site’s file system. However, the library “saltcc.js” can reside in any valid subdirectory.

Sections 4, 5, and 6 of Figure 2 show a sample of the markup language you might use to add telephony call-control features to a Web site. (Note that, in the HTML code samples, the source URL is shown as a static .html file. In practice, you would most likely use a server-side scripting technology such as Java* servlets, Perl, CGI scripts, Microsoft Active Server Pages*, and so on.)

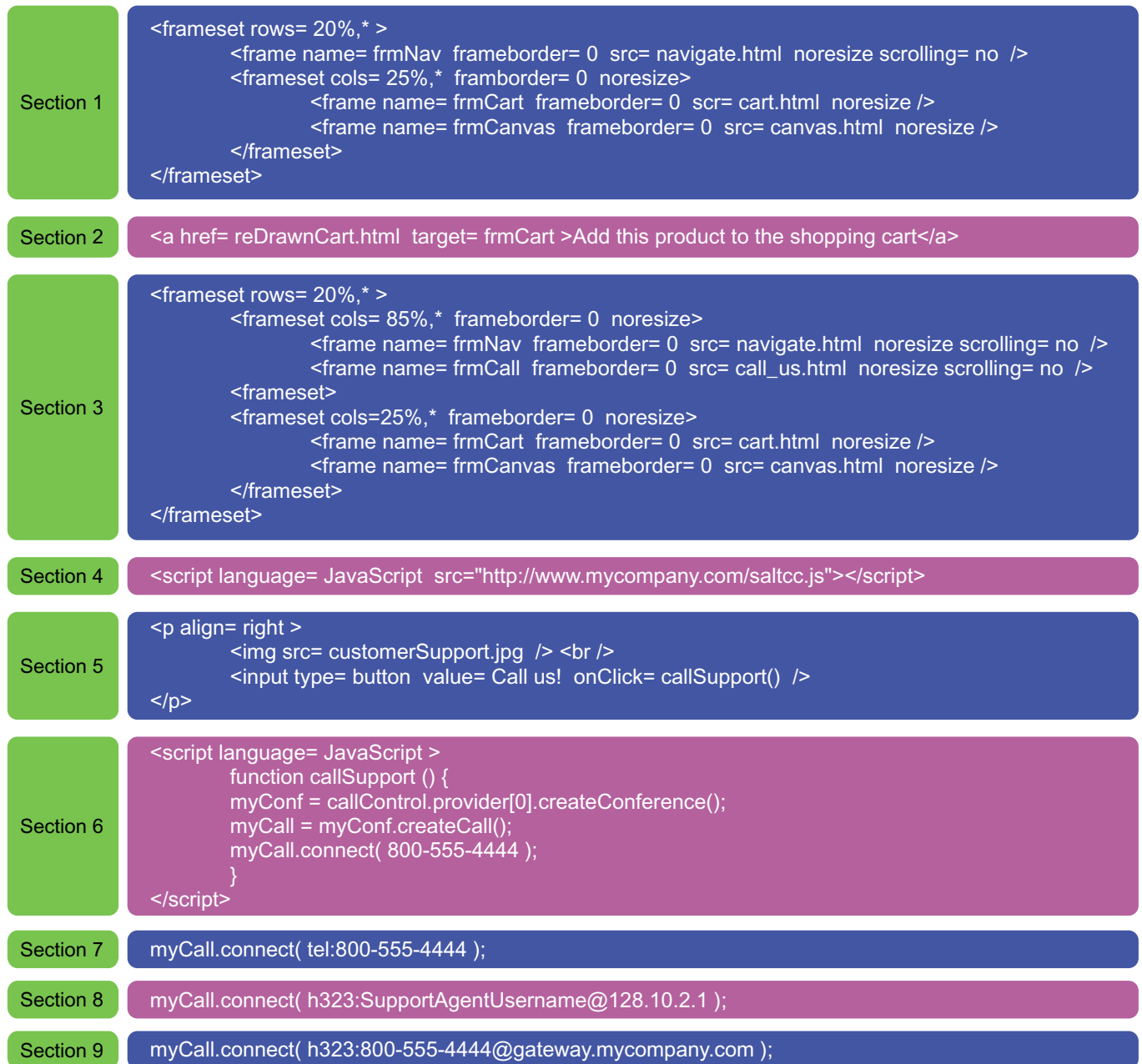


Figure 2. The code excerpts shown here are examples of call-control features that can be added via a standard markup language.

Section 5 of Figure 2 shows how to add a “click to dial” button to the site. Section 6 shows an example of adding the JavaScript function “callSupport.” The “callSupport” function will be called when the user clicks the button to make the telephone call.

Cable, DSL, Wireless

The telephone call is placed directly over the Internet using a technology called Voice over Internet Protocol (VoIP). This allows calls to be made over virtually any type of Internet connection, including cable, DSL, wireless, or 56K phone-line modem. It allows the user to stay online throughout the call and avoids long-distance charges, which can be particularly important for worldwide commerce.

VoIP technology is available now on Microsoft Windows^{*} operating systems. Many provider programs offer VoIP technology for the different communication methods. For example, Intel offers Internet Phone*, Microsoft offers NetMeeting* and MSN Messenger*, and so on. The provider program you can use will depend on the version of the call-control library you install on your Web server. However, you can install and use more than one provider program on your PC.

Phone Number Format

The provider program may also determine the format of the telephone number used to make the call. The number must always be in the format of an Internet URL (uniform resource locator). This is similar to an “http://” string. However, because VoIP uses different protocols than HTTP, telephone URLs begin with either “tel:”, “h323:”, or “sip:”.

The protocol you should use will depend on:

- Whether you want your support personnel to answer calls using PCs or telephones.
- The types of servers and server configurations the callers have on their PCs.

For example, the “tel:” URL is the easiest and most obvious to use. To use a “tel:” URL in the markup language, you would change the “myCall.connect” number (“800-555-4444”) in section 6 of Figure 2 to the URL (“tel:800-555-4444”) shown in section 7.

Unfortunately, the “tel:” URL presumes that the user’s PC is preconfigured to contact a special server called a gatekeeper/gateway. A gatekeeper/gateway server converts the VoIP call into a telephone call to or from the usual, physical telephone. It is highly unlikely that a home user will have a gatekeeper/gateway server for their PC.

Both the “h323:” and “sip:” URLs let you specify a destination and an IP address like an e-mail address. Section 8 of Figure 2 shows an example of an IP address that is the URL of a customer support agent’s PC. Section 9 shows how to specify an actual telephone number and the IP address of a VoIP gateway.

Once you know the telephone protocol to use, you are ready to start implementing the telephony call-control features.

Dropped Calls

A common problem seen when telephony features are first added to sites is that of having the phone call cut off when the user clicks on a different Web feature.

Remember that when an HTML page is loaded into the browser, all objects on the page (text, buttons, images) are instantiated in memory. This also holds true for any JavaScript code on the page, including any SALT call-control libraries or developer-written functions. When a new page is loaded, all objects on the previous page are “destroyed” or rather, de-allocated from memory. As far as the browser is concerned, they cease to exist. One of the side effects of this is that any telephony calls are disconnected.

There are several ways to fix this problem so that the phone call continues even when the user clicks to a new page. One solution is using window.open() to spawn a new window dedicated to the phone call. Alternatively, frames provide an excellent solution for making calls within the existing window.

You’ve Been Framed

In HTML, frames are grouped into a collection called a frameset. (Figure 2, section 1 shows the sample definition of a typical frame for a Web site.) A standard e-Commerce site has several frames. The navigation frame might be a menu that lists different product categories or site features. The main frame might be a display of the products available for sale. The shopping-cart frame would probably list the products that the user has already selected.

Usually, when a link is clicked within a frame, only the contents of that frame are altered, and other frames are not affected. However, sometimes, you want a link in one frame to change the contents of another frame. For example, if a user clicks on a feature titled, “Add this product to the shopping cart,” in one frame, the shopping-cart frame should be updated and redrawn with the new information.

To add telephony features to the Web site, you should first add a fourth frame—a call-control frame—to your site's frameset. The SALT call-control library and the JavaScript functions for call management will then be stored in the HTML page that is displayed in that frame.

For all frames, you use the `target=` attribute in link definitions (refer to section 2 of Figure 2). The `target=` attribute lets you make sure that, after a user activity, the only frame that gets redrawn is the one named in the target attribute (for example, `frmCart`), rather than the frame in which the link resides. In this way, any or all of the other frames can be redrawn without affecting the call-control frame. Call-control objects and their associated state can be persistent, regardless of other site activity; and active calls will not be unintentionally disconnected.

(Note that you can also use the `target=` attribute in the call-control frame to allow JavaScript functions in that frame to cause changes in other frames.)

Using a separate frame for call control will help you offer robust telephony features to your customers. After all, telephone features on e-Business sites should help relieve user frustration, not increase it.

Summary

This article describes only one of the many ways you can use SALT call-control capabilities to integrate innovative communications features within a Web site. A draft version of the SALT specification is available now, and the fully approved version of the specification should be available in early July 2002. Because SALT tools and software development kits are also already available, you can start experimenting now with both call control and speech I/O. By starting now, you can get a jump on business and technology plans that include the use of telephony features for online customers.

More Info

This article is second in a series. The first article, titled "Telephony Call Control Now Available for HTML, XHTML, and XML" appeared in the June 2002 issue of Intel Developer Update magazine. The most recent draft of the SALT specification is available at the SALT Forum Web site. The draft specification describes the new SALT tags and the ECMAScript call-control object methodology.

The "Speech Application Language Tags (SALT) Technical White Paper" is also available online at the forum's site. The white paper describes design principles, the three main (top-level) SALT elements, event handling, telephony capabilities for call control, and the flow of dialog in voice-only scenarios.

Intel's "SALT Call Control Technology Preview" is now available on the Intel® Developer Services site. The preview includes open-source libraries, example code, tools, and documentation for developers to learn more about SALT Call Control and develop proof-of-concept pages. Microsoft's beta-release SALT tools are also currently available at Microsoft's speech.NET Web site.

Author Bios

Glen Shires is currently director of telephony media servers in the Intel Communications Group. Glen has worked in a variety of software and hardware roles at Intel, from applications engineering for the Intel® 386™ microprocessor to graphics, speech, and telephony systems development. Today, he manages speech technologies on CT Media and has been a technical contributor to the SALT specification since the formation of the SALT Forum in 2001. He has been awarded three patents and has five patents pending. Glen received his B.S. and M.S. in electrical and computer engineering from the University of Wisconsin-Madison. For his master thesis, he designed and built a computer speech recognition system.

Jim Trethewey has been with Intel for 19 years, and is currently a staff software engineer for Intel Labs. Jim has worked in numerous capacities at Intel, including hardware design, operating system and applications software design, and technical marketing. Today, he is a technical contributor to XML-related specifications in the SALT Forum and in the World Wide Web Consortium (W3C). Jim developed the reference implementation of SALT Call Control, which is now available in Intel's Technology Preview. Jim received his B.S. in computer science from Oregon State University in 1983. He has been awarded two patents, and has three patents pending.

—End of Intel Developer Update Magazine Article—