

Bringing Security to Macromedia® Director®

By Gabrielle Dara Krim

A White Paper from Integration New Media



Integration New Media, Inc.

1600 Rene-Levesque Blvd W, Suite 900
Montreal, QC H3H 1P9
Canada

Tel.: +1 514 871-1333
Fax: +1 514 871 9251

www.INM.com

Security is a Sign of Maturity

E-commerce, pay-per-use online services, user-authentication and tracking for e-learning, online gaming, contests... What do all of these applications have in common? The need for secure transfer of encrypted data between client workstations and server applications. According to a study by **Gartner Consulting**, the growing concern for Internet security parallels the evolution of e-business. In the earliest days of Internet development, the emphasis was on distributing content over the web and making it available to anyone. Now, as the Internet matures, clients are more concerned with ensuring that their assets, both monetary and intellectual, are protected from those who may commit fraud or abuse them. This is why more and more developers have been looking for security solutions.

The Technological Challenge

Secure HTTP, or Hypertext Transfer Protocol with Secure Sockets Layer (**SSL**), is the current standard for transmitting secure information over the Internet. The SSL security protocol, developed by Netscape® Communications® Corporation, provides data encryption, server authentication, message integrity, and optional client authentication for a TCP/IP connection. It uses public key cryptography, a technique that uses a pair of asymmetric keys for encryption and decryption. Data is encrypted using a public key, but can only be decrypted using a private key. URLs that use Secure HTTP begin with `https://`.

Out of the box, Director®-based applications can only offer Secure HTTP connections through Shockwave® within a browser. There is no way to ensure secure communications from a projector. Likewise, there is no efficient way to debug Secure HTTP communications in Director's authoring environment.

Common Workarounds and Consequences

There have been few alternatives for Director users:

- Spend many hours trying to overcome the problem by combining encryption Xtras with NetLingo® and developing custom encryption/decryption scripts on the server - resulting in time and money wasted, as well as frustration for developers and their clients.
- Give up on security, thus jeopardizing the integrity of data and leading to potential disasters - e-learners could cheat, contests could be tricked, confidential information could fall into the wrong hands.
- Deliver the project in Shockwave running within a browser - even if the project is not suited for that kind of delivery.
- Avoid using Director - replace with a non-multimedia and more restrictive development tool that incurs greater development expenses.

Integration New Media's SecureNet Xtra

At Integration New Media (INM), we understood developers' needs for secure Internet communication from both within and outside the browser environment. We built SecureNet Xtra with those needs in mind.

SecureNet Xtra provides nearly all the network communication methods of NetLingo - secure HTTP queries, posting form data to CGI or ASP scripts, downloading files -- and these operations can be performed from projectors, Shockwave and even in Director's authoring environment. Developers can save time -- even on Shockwave applications which could use the browser's own HTTPS implementation when deployed-- with the ability to test secure connections directly from the authoring mode. SecureNet Xtra piggy-backs on Internet Explorer's® `wininet.dll` to provide the SSL encryption/decryption protocol for secure HTTP communications.

This table provides a comparison of some of the commands and functions used by standard Net Lingo and the SecureNet Xtra.

Net Lingo methods	SecureNet Xtra methods and behaviors	Differences in SecureNet Xtra implementation from Net Lingo
<code>DownloadNetThing</code>	<code>snxDownloadNetThing</code> <code>snbDownloadNetThing</code>	Only HTTP and HTTPS; not FTP. Allows downloading files from a secure HTTPS server from a projector or Director movie
<code>GetNetText</code>	<code>snxGetNetText</code> <code>snbGetNetText</code>	Only HTTP and HTTPS; not FTP. Does not support multiple OS character set. Use of <code>netID</code> parameter is mandatory. Allows retrieval of text from a secure HTTPS server from a projector or Director movie
<code>PostNetText</code>	<code>snxPostNetText</code> <code>snbForm.Field</code> <code>snbForm.Submit</code>	Only HTTP and HTTPS; not FTP. Does not support multiple OS character set. Use of <code>netID</code> parameter is mandatory. Allows posting to a secure HTTPS server from a projector or Director movie
<code>GetStreamStatus</code>	<code>snxGetStreamStatus</code>	Does not support syntax with an URL passed as a parameter, only <code>netID</code>
<code>NetAbort</code>	<code>snxNetAbort</code>	Does not support syntax with an URL passed as a parameter, only <code>netID</code>
<code>NetDone</code>	<code>snxNetDone</code>	Use of <code>netID</code> parameter is mandatory
<code>NetError</code>	<code>snxNetError</code> <code>snsNetError</code> <code>snsDisplayError</code>	<code>snxNetError</code> doesn't support the no parameter syntax <code>snsDisplayError</code> and <code>snsNetError</code> allow for multiple syntaxes
<code>NetTextResult</code>	<code>snxNetTextResult</code> <code>snbGetNetText</code> <code>snbPostNetText</code>	Use of <code>netID</code> parameter is mandatory. The Behaviors allow you to optionally display the returned result directly to a field or text member.
<code>ProxyServer</code>	<code>snxProxyServer</code>	SecureNet Xtra allows for additional syntaxes, with the ability to accept username and password

When Not to Use SecureNet Xtra

In most situations, you can safely use SecureNet Xtra in place of NetLingo for your network communications. However, there are a few situations in which SecureNet Xtra would not be desirable:

- To transfer files to or from an FTP server (as of the writing of this article, FTP is not supported).
- For Authorware Mac OS X users (as of the writing of this paper, the Mac OS X version is not available for Authorware).
- If the user is running Director on a shift-JIS (Japanese) system.

How to Implement Network Operations Using SecureNet Xtra

There are two ways to implement network operations with SecureNet Xtra: using behaviors, and using Lingo methods. The rest of this article describes how to implement a sample project called **First Steps**, using both approaches. The behavior version of this sample is available from the SecureNet Xtra demo page: <http://www.integrationnewmedia.com/products/snxdirector/demos/>

INM's SecureNet Xtra

integration
new media
MADE WITH
Macromedia

Instructions

Type in the following credit card information and click Submit.
Card holder's name: *Your Name*
Credit card number: 4500-7777-7777-7777
Expiry date: 2004

TicketScalper
Secure Online Payment Form

Cardholder's name:

Credit Card Number:

Expiry date:

URL:

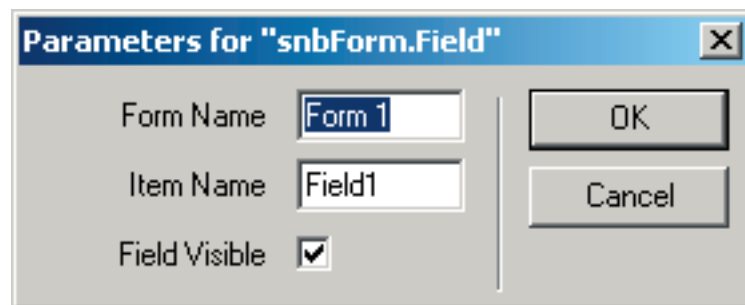
Results:

First Steps v1.0 ©2002, Integration New Media, Inc.

The context is an e-purchasing application that accepts credit card information and sends it to an ASP script on a server implementing a Secure HTTP connection. The basic operation you need to implement is the standard HTTP POST, which most Internet forms employ when the user clicks a Submit button.

Using SecureNet Xtra Behaviors

The SecureNet Xtra Behaviors needed for this operation are **snbForm.Field** and **snbForm.Submit**. **snbForm.Field** associates the contents of a Director field with a specific variable on the server-side script. To properly submit the credit card information, you need to drag the **snbForm.Field** behavior over each field that contains data that the server-side ASP script requires, such as **Last Name, First Name, Card Number, Expiry Date, Amount**, etc. When you drag the behavior onto a text field on the stage, this dialog box opens.



For details on these parameters, download:

The **SecureNet Xtra User Manual** from:

<http://www.integrationnewmedia.com/support/snxdirector/manuals/>

The **First Steps tutorial** from:

<http://www.integrationnewmedia.com/products/snxdirector/demos/>

Once you have identified all the fields whose data will be sent, drag the **snbForm.Submit** behavior to a button on the stage that will trigger the network operation when clicked. This dialog box appears:

From this box, you can define most of the parameters by simply selecting from drop-down menus. The URL to which you are posting data is stored within a Director cast member, as is the result returned, if applicable. Note the proxy server settings in the lower half of the dialog box. These settings allow your application to communicate through an end-user's proxy server if there is one. See the SecureNet Xtra User Manual for details.

With SecureNet Xtra behaviors, the communication is synchronous - that is, Director waits until the network operation has completed before processing any other commands. The user will fill in his credit card information, click a Submit button, and wait to receive some notification, which is either a response sent by the remote URL or something else you choose to do in your application.

Using SecureNet Xtra Methods

Now let's look at how to implement the same kind of example using the Lingo® methods. In Lingo, you have the option of coding SecureNet Xtra communications in either synchronous or asynchronous modes.

As with the behaviors, you would likely have the data to be sent stored in fields on the stage. You would then send the data using the method `snxPostNetText` within the `mouseUp` handler of a **Submit** button. Between each network operation, it's good practice to call one of the error handling methods, which are supplied in a script within the Behaviors Library.

Synchronous Example:

```

-- send the variables to the server-side script
NetID = snxPostNetText
(https://www.integrationnewmedia.com/creditcard.asp, [#name:myName,
#cardnumber:myNumber, #expires: myExpires])
-- if an error occurred, display the error directly in an alert box and don't
continue
if (snsDisplayError (NetID, 1)) then
  exit
end if

-- the POST command was initiated successfully
-- now check for completion
-- repeat/while loop is used to implement synchronous communication
-- and prevent Director from processing other commands
repeat while (NOT snxNetDone (NetID))
  -- get the status of the operation
  IStatus = snxGetStreamStatus (NetID)
  pctDone = (IStatus.BytesSoFar / IStatus.BytesTotal) * 100
  -- display a progress indicator while the user waits
  ...
end repeat
-- check for result status
if (snsDisplayError (NetID, 1)) then
  -- don't continue if errors occurred
  exit
end if
-- display any response from the server script
member ("Result").text = snxNetTextResult (NetID)

```

If you choose to implement the communication synchronously, with the `repeat` loop, the user cannot interact with Director until the operation has completed. This is usually desirable for network operations that are reasonably quick. If, however, you want the user to be able to send multiple network communications without waiting for a response between each, or you want to display an animation while the user waits, you may use the asynchronous mode.

Asynchronous Example:

```

on mouseUp
  -- initialize variables to send here
  -- send the variables to the server-side script
  gNetID = snxPostNetText
  (https://www.integrationnewmedia.com/creditcard.asp, [#name:myName,
  #cardnumber:myNumber, #expires: myExpires])
  -- if an error occurred, display the error directly in an alert box and don't
  continue
  if (snsDisplayError(gNetID, 1)) then
    exit
  end if
  -- the POST command was initiated successfully
end mouseUp

-- now check for completion
on exitFrame
  -- get the status of the operation
  IStatus = snxGetStreamStatus(NetID)
  pctDone = (IStatus.BytesSoFar/IStatus.BytesTotal) * 100
  -- display a progress indicator while the user waits
  ...

  -- check for completion
  if (snxNetDone (gNetID)) then
    if (snsDisplayError (gNetID, 1)) then
      -- don't continue if errors occurred
      exit
    end if

    -- display any response from the server script
    member ("Result").text = snxNetTextResult (gNetID)

    -- go to a specific frame
    -
  else
    go to the frame
  end if
end exitFrame

```

Note that with the asynchronous mode, you must use a global variable: `gNetID` instead of `NetID`.

Conclusion

Internet-based applications are here to stay. Even applications delivered on CD-ROM now offer some dynamic, web-based content. Therefore, security has become an important concern for project sponsors, end-users and developers alike. Project sponsors, in particular, who fund development projects and expect a return on their investments, are driving the trend, demanding and expecting certain measures of security for their Internet-based applications. Moreover, they are willing to pay extra for applications that can offer security.

The Gartner Consulting paper lists the “ability to provide authentication and authorization functionality as part of the defined security solution” as one of the key criteria in evaluating a software vendor. They conclude by stating, “a vendor who is able to offer these [secure] software tools and integration services can truly help their customers leverage the power of the Internet for e-business.”

The bottom line for Director developers is that we all have to stay abreast of current security concerns or risk being left behind. With SecureNet Xtra, we now have a new option for secure application development that doesn't compromise the multimedia experience that has earned Director its worldwide recognition.

System requirements

As of the writing of this white paper, SecureNet Xtra version 1.3 is available for Director (Win/Mac) and version 1.2.1 is available for Authorware (Windows only).

- Macromedia Director version 7.02 + or Macromedia Authorware version 4.0 +
- Windows: Windows 95/98/NT4/Me/2000/XP.
- Microsoft Internet Explorer 4.x and later (or at least the corresponding wininet.dll library).
- Mac OS 9.x (with URL Access Manager version 2.4.1 or later)
- Mac OS X version 10.2.x (Director only)

Product Info

- SecureNet Xtra is available from the INM store for \$US 199.
www.IntegrationNewMedia.com/store/
- The Shockwave download weighs less than 80K.
- Installation is automatic, Verisign-approved, and safe for end-users.

Visit www.SecureNetXtra.com for more information and a free trial download.

References:

www.INM.com

To learn more about Secure HTTP and the Secure Sockets Layer (SSL) protocol, visit: <http://wp.netscape.com/security/techbriefs/ssl.html>

Gartner Consulting's white paper, "The Evolution of e-Business Security Requirements", prepared for Verisign, Inc.
<http://www.verisign.com/resources/wp/authentication/eBusinessSecurity.pdf>

Gabrielle Krim has a B.A. in Computer Science from Cornell University and a M.A. in Educational Technology from Concordia University. She has over 10 years experience in the industry, as a software developer and multimedia author for database and e-learning projects. She has been working for INM as Instructional Designer and Project Manager since March 2002.