

## *How to use the Instant STLinks package with an LTPA Token*

---

LTPA tokens are created when a user signs on to Domino or Sametime servers that are configured for Single Sign-On (SSO). They are stored as a cookie in the user's browser cache, and contain the information the server needs to re-authenticate the user "on demand", thus allowing a SSO capability. The Instant STLinks package can take advantage of this capability to allow logon within a portal, for instance, and not require a separate logon for the Instant STLinks Sametime logon.

There are two basic things required to enable this ability:

1. The LTPA token .
2. The user's name.

While it is true that the LTPA token technically contains the user's name, extracting it is beyond the scope of this document. Instead, this document will focus on using the Instant STLinks with a Domino server. Since the Domino server already has the user's login information, requesting it from the server in the form of code on a Domino page is the simplest solution.

To retrieve cookies using JavaScript, you can use the following function

```
<script>
function readCookie(name) {
    var nameEQ = name + "=";
    var ca =
    document.cookie.split(';');
    for(var i=0;i < ca.length;i++) {
        var c = ca[i];
        while (c.charAt(0)==' ') c = c.substring(1,c.length);
        if (c.indexOf(nameEQ) == 0) return c.substring(nameEQ.length,c.length);
    }
    return null;
}
</script>
```

To retrieve the LTPA token (in this example, we will assume it is called "LtpaToken"), use the following code element in the HTMLHead event of a Page or Form element in a Domino application:

```
"<script>
    writeSTLinksApplet(\"\" + @LowerCase(@ReplaceSubstring(@UserName; "/" ; ",")) + "\",
readCookie(\"LtpaToken\"), true);
</script>"
```

Note the double-quotes around the code. This is a hard-coded JavaScript event, placed in the HTMLHead section, so that the Domino server will pass the code through to the browser unmolested. Also note that the second piece of the equation, retrieving the user's name, is also present in this small piece of code. In this instance, the example shows how to retrieve the user's name, and tweak it to pull from an LDAP directory in the correct format (hence the @LowerCase and @ReplaceSubstring elements, which remove the slashes of a standard Domino name and replace them with commas for a standard LDAP name).