

Impersonate progromatically persist through session or just page?

Source:

<http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.framework.aspnet.security/2002-10/1316.html>

From: Patrich Lynch (patldev@rampros.com)

Date: 10/02/02

From: patldev@rampros.com (Patrich Lynch)

Date: 2 Oct 2002 10:53:19 -0700

Regarding impersonating a user (see the below code from newgroup). A quick question. In a ASP.NET Internet application, I want to have users login in against a database. In their user record in the database, they have a security level that will correspond to a particular OS user. When a user first hit's the site, they are logged on as anonymous using the Identity provided in the web.config file. When sucessfully logged in against the DB, the user OS account will/may be switched (impersonated) to the appriate account. In my senario, there would be no undoImpersonation() unless/until they logged out. My question: does the impersonation carry throughout the session/process (i.e. from request to request) or does it only apply to the currently executing request? TIA

Patrick Lynch
Web Developer
patldevNOSPAM@rampros.com
Cell: 510.381.5329

Impersonate a Specific user wherever required within the code

----- When
you want to impersonate a specific user only during the execution of certain piece of code, you can use the following mechanism to do that.

C#

```
[DllImport("advapi32.dll", CharSet=CharSet.Auto)]
public static extern int LogonUser(String lpszUserName,
    String lpszDomain,
    String lpszPassword,
    int dwLogonType,
    int dwLogonProvider,
    ref IntPtr phToken);
```

microsoft.public.dotnet.framework.aspnet.security: Impersonate progromatically persist through session or just page

```
[DllImport("ADVAPI32.DLL")]
public static extern int RevertToSelf();

[DllImport("ADVAPI32.DLL")]
public static extern int ImpersonateLoggedOnUser(IntPtr phToken);

if impersonateValidUser("accountname", "Domainname", "password") {

    //Your code that runs under the security context of a specific
user
goes here.....

    undoImpersonation();
}
else {
    //your impersonation failed. So, have a failsafe mechanism here
}

public boolean impersonateValidUser( String name, String domain,
String
passwd) {

    IntPtr tok = IntPtr.Zero;
    public const int LOGON32_LOGON_INTERACTIVE = 2;
    public const int LOGON32_PROVIDER_DEFAULT = 0;

    int result = NativeMethods.LogonUser(name, domain, passwd,
        LOGON32_LOGON_INTERACTIVE,
        LOGON32_PROVIDER_DEFAULT,
        ref tok);
    impersonateValidUser = "False";
    if result!=0 {
    int result1 = ImpersonateLoggedOnUser(tok);
    if result1 != 0 return "True";
    }
}

public void undoImpersonation(){
RevertToSelf();
}
```