

## Properly removing the domain prefix requirement from RD Web Access 2012 R2

In the first post in the [Customizing the RD Web Access 2012 R2 interface series](#) I added a section that describes how to remove the Domain prefix in the login.aspx page.

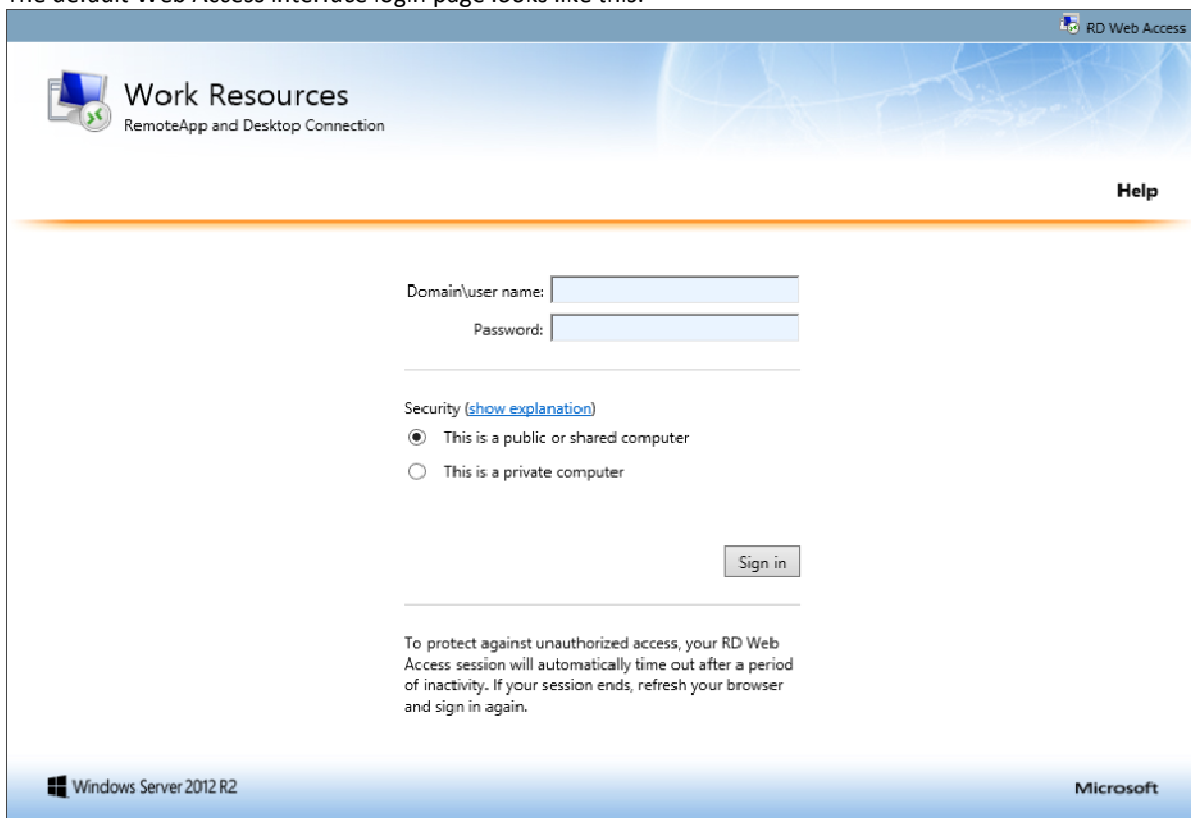
Initially my customization only functioned for Domain joined machines, so the post required an update. The updated customization fixed the initial problem: non Domain joined machines accessing the RD Web Access pages can now successfully log on using their username only, without prefixing it with the Domain Netbios name. Until someone tested the Expired Password or Change Password functionality... It turns out that this page uses different code to change the password, and this code requires a Domain prefix, or UPN.

I decided not to post another update, but instead dedicate a full post to this issue, and the solution to it. If you now look at the original post you'll see that I refer to this post when this customization is discussed.

So here goes, a full solution to properly remove the Domain Prefix requirement from the RD Web Access pages.

As always, backup the files in %windir%\Web\RDWeb\Pages, just in case.. I'm using the same environment I used in the original post, so for more info on that read that post.

The default Web Access interface login page looks like this:



As you can see, the interface by default expects the user to enter the username in the NT Account format, e.g. Domain\user name.

### Removing the domain prefix requirement from the login page

If you publish the Web Access interface for a single domain infrastructure or if you want to define a default domain to logon to, you might want to consider to let the user just enter the username instead of the NT Account.

By default the Web Access application won't let you do this:

Domain\user name:

Password:

**You must enter a valid domain name.**

Besides, even if it would, it still shows "Domain\user name:" on the label. So we need to fix two things: the text on the label, and some code to accept just the username.

Open "login.aspx" and move to line number 19:

```
const string L_DomainUserNameLabel_Text = "Domain\\user name:";
```

This line holds the text for the label.

Change this line to:

```
const string L_DomainUserNameLabel_Text = "User name:";
```

Or replace "User name:" with a custom text you prefer.

Save the file.

This changed the label text to just ask for a username, and not to enter a domain prefix.

Open "webscripts-domain.js" and move to line number 7:

```
var strDomainName = null;
```

This line tells the rest of the code that by default there's no domain available and it needs to be extracted from whatever the user entered in the fields in the login page.

Change this line to:

```
var strDomainName = "ITW";
```

And of course replace "ITW" in this example with your own NETBIOS name for the domain.

Now move to line number 75 and insert the following code just before line 75:

```
if (-1 == strDomainUserName.indexOf("\\"))
{
    if (-1 == strDomainUserName.indexOf("@"))
    {
        strDomainUserName = strDomainName+"\\"+strDomainUserName;
    }
}
```

The result must look like this:

```
try
{
    var iWorkspaceVersion = GetWorkspaceObjectVersion();
    var objWorkspace = GetWorkspaceObject();

    if (-1 == strDomainUserName.indexOf("\\") )
    {
        if (-1 == strDomainUserName.indexOf("@") )
        {
            strDomainUserName = strDomainName+"\\"+strDomainUserName;
        }
    }

    if (iWorkspaceVersion >= 2)
```

Now save the file.

Open "renderscripts.js".

As the very first line, insert the following code:

```
var strDomainName = "ITW";
```

Replace "ITW" with your own NETBIOS domain name.

The result of adding this line must look like this:

```
var strDomainName = "ITW";
var helpPopup = null;
var strTSWACookieName = "TSWAAuthClientSideCookie";
var strTransparentGif = "../images/clear.gif";
```

Now move to line 334 and insert the following code there:

```
if ( (-1 == strDomainUserName.indexOf("\\")) && (-1 == strDomainUserName.indexOf("@")) )
```

```
{
strDomainUserName = strDomainName+"\\"+strDomainUserName;
}
```

The result must look like this:

```
if ( strDomainUserName.ToLowerCase() == strLoggedInDomainUserName.ToLowerCase() ) {
    if ( ( -1 == strDomainUserName.IndexOf("\\") ) && ( -1 == strDomainUserName.IndexOf("@") ) )
    {
        strDomainUserName = strDomainName+"\\"+strDomainUserName;
    }

    if (GetActiveXSSOMode()) {
        try {
```

Now save this file as well.

### Removing the domain prefix requirement from the password page

If you decided to allow users to change their password, the user can click a link to open the password.aspx page when they log on, or when the pages detect the password has expired.

If you want to remove the Domain Prefix from the RD Web Access interface, you need to do some more editing. Open "password.aspx" and move to line number 19:

```
const string L_DomainUserNameLabel_Text = "Domain\\user name:";
```

This line, again, holds the text for the label.

Change this line to:

```
const string L_DomainUserNameLabel_Text = "User name:";
```

Or replace "User name:" with a custom text you prefer.

On line 36 add a new constant:

```
const string L_DefaultDomain = "ITW";
```

Again, replace "ITW" with your own NETBIOS domain name.

Goto line 112:

```
DomainUserName.Value = SecurityElement.Escape(objQueryString["UserName"]);
```

Add the following code after this line:

```
ShortUserName.Value = DomainUserName.Value;
if ( !(DomainUserName.Value.Contains("\\") )
{
    DomainUserName.Value = L_DefaultDomain+"\\"+DomainUserName.Value;
}
```

The result must look like this:

```
if ( objQueryString["UserName"] != null )
{
    DomainUserName.Value = SecurityElement.Escape(objQueryString["UserName"]);
    ShortUserName.Value = DomainUserName.Value;
    if ( !(DomainUserName.Value.Contains("\\") )
    {
        DomainUserName.Value = L_DefaultDomain+"\\"+DomainUserName.Value;
    }
}
```

To prevent the UserName box to become empty if the user fails to enter correct values, this next piece is kind of nasty, but necessary ;)

Find line 174:

```
<form id="FrmLogin" name="FrmLogin" action="password.aspx" method="post">
```

And change that line into

```
<form id="FrmLogin" name="FrmLogin" action="password.aspx?UserName=<%=ShortUserName.Value%>" method="post">
```

The result of that must look like this:

```
<HTMLMainContent>
<form id="FrmLogin" name="FrmLogin" action="password.aspx?UserName=<%=ShortUserName.Value%>" method="post">
  <table width="350" border="0" align="center" cellpadding="0" cellspacing="0">
```

Really really dirty, but gets the job done.

Next, find line 189:

```
<input id="DomainUserName" name="DomainUserName" type="text" class="textIn
```

Delete this line, and replace it with the following code:

```
<input id="ShortUserName" name="ShortUserName" type="text" class="textInputField" runat="server" size="25"
autocomplete="off" disabled />
<input id="DomainUserName" name="DomainUserName" type="hidden" runat="server" />
```

The result must look like this:

```
<tr>
  <td width="180" align="right"><%=L_DomainUserNameLabel_Text%></td>
  <td width="7"></td>
  <td align="right">
    <input id="ShortUserName" name="ShortUserName" type="text" class="textInputField" runat="serve
    <input id="DomainUserName" name="DomainUserName" type="hidden" runat="server" />
  </td>
</tr>
```

Save the file.

The password.aspx file is only meant to be accessed if the user is already authenticated. The password.aspx file is not meant to be used to allow users to change their password without being authenticated first. Examples of this include adding a link on the login.aspx page to password.aspx, or simply using the aspx files to allow domain users to just change their password, not using the RD Web Access at all.

If you do want to add a link to password.aspx to allow password change, I suggest you add a link on the toolbar. I describe how to do that in the Customization series.

To conclude:

This added "ITW" as the default authentication domain in both the login page as the password change page. Nothing changed in the rest of the code, so if your Web Access is intended for multiple domains, the user can still enter "CHILDDOMAIN\user name" or "TRUSTEDDOMAIN\user name" or even "ITW\user name" if the user wanted to do that. We've also not destroyed the possibility to logon using UPN instead of NT Account logon.

These changes are instant, there's no need to restart IIS. Just (re)load the Web Access page and test the changes.

Unfortunately, with this customization you need to enter the NETBIOS name hardcoded in three different files. In a future post, which will cover a full customization package with application settings in IIS I will show how to eliminate this and make Domain NETBIOS name an application setting which works across the complete RD Web Access interface.

Until next time,