

## Adding the current user's Active Directory displayname to RD Web Access 2012R2

**Update (October 24th, 2014): I modified the code slightly to support special characters in the user's displayname attribute.**

This is another addition to the "Customize RD Web Access 2012 R2" series. You can start this series [here](#) in case you want to read up or see what else I customized in this series.

This post is inspired by comments in the series, in which readers asked if it was possible to show the logged in user's displayname as it appears in Active Directory.

I did a little digging in the code and found a way to accomplish this.

I'm not going to detail the setup I have here. If you're interested I suggest you read up on the series.

Logging into the RD Web Access shows this:



This is actually the top of the Default.aspx page.

Just basic stuff here, nothing customized, it shows a basic navigation bar. I'm going to show you how you can add the displayname right after the "Sign out" button on the navigation bar.

The files that make up RD Web Access are in C:\Windows\Web\RDWeb\Pages so make a copy of that folder structure just to make sure we have a backup should we break something.

We are going to use native Windows code from an assembly that is already present on the server that holds the RD Web Access role, but is not known to the RD Web Access application yet. So the first step is to tell the application that we are going to add code.

Open web.config from C:\Windows\Web\RDWeb\Pages in an editor.

Find line 52:

```
<system.web>
```

And add the following code right after that:

```
<compilation defaultLanguage="c#" debug="true">
  <assemblies>
    <add assembly="System.DirectoryServices, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a" />
  </assemblies>
</compilation>
```

The result must look like this:

```
<system.web>
  <compilation defaultLanguage="c#" debug="true">
    <assemblies>
      <add assembly="System.DirectoryServices, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />
    </assemblies>
  </compilation>
<!--
```

This tells the RD Web Access application that we need extra code. The extra code is in the .NET Assembly, and specifically in System.DirectoryServices.dll.

Save and close web.config, we don't need it anymore.

Now open Default.aspx in an editor, which is in C:\Windows\Web\RDWeb\Pages\en-us.

Find line 225:

```
</script>
```

Insert the following code before line 225:

```

private static string GetDisplayName(string strUserName)
{
    string strLDAPPath = "LDAP://dc=itw,dc=test";
    string strFilter = string.Empty;

    if(strUserName.Contains("\\")){strUserName = strUserName.Substring(1 + strUserName.IndexOf("\\"));}
    strFilter = "(SAMAccountName=" + strUserName + ")";
    if(strUserName.Contains("@")){strFilter = "(UserPrincipalName=" + strUserName + ")";}

    System.DirectoryServices.DirectoryEntry de = new System.DirectoryServices.DirectoryEntry(strLDAPPath);
    System.DirectoryServices.DirectorySearcher ds = new System.DirectoryServices.DirectorySearcher(de);
    ds.Filter = strFilter;

    ds.PropertiesToLoad.Add("DisplayNameSystem.DirectoryServices.SearchResultCollection results =
ds.FindAll());

    return (results != null && results.Count > 0) ?
Uri.EscapeDataString(results[0].Properties["DisplayName"][0].ToString()) : string.Empty;
}

```

Remember to replace “LDAP://dc=itw,dc=test” with your own domain name. So for example, if your domain name is “contoso.local”, this code would be “LDAP://dc=contoso,dc=local”.

The result must look like this (I have wordwrap on in my editor):

```

private static string GetDisplayName(string strUserName)
{
    string strLDAPPath = "LDAP://dc=itw,dc=test";
    string strFilter = string.Empty;

    if(strUserName.Contains("\\")){strUserName = strUserName.Substring(1 + strUserName.IndexOf("\\"));}
    strFilter = "(SAMAccountName=" + strUserName + ")";
    if(strUserName.Contains("@")){strFilter = "(UserPrincipalName=" + strUserName + ")";}

    System.DirectoryServices.DirectoryEntry de = new System.DirectoryServices.DirectoryEntry(strLDAPPath);
    System.DirectoryServices.DirectorySearcher ds = new System.DirectoryServices.DirectorySearcher(de);
    ds.Filter = strFilter;
    ds.PropertiesToLoad.Add("DisplayName");
    System.DirectoryServices.SearchResultCollection results = ds.FindAll();

    return (results != null && results.Count > 0) ? Uri.EscapeDataString(results[0].Properties["DisplayName"][0].
ToString()) : string.Empty;
}
</script>

```

This shows line 225-242 in my file.

Now find line 249:

```
baseurl="<%=SecurityElement.Escape(baseUrl.AbsoluteUri) %>"
```

And add the following line of code right after that:

```
userdisplayname="<%=GetDisplayName(strDomainUserName)%>"
```

The result must look like this:

```

<RDWAPage
    helpurl="<%=sHelpSourceServer%>"
    domainuser="<%=SecurityElement.Escape(strDomainUserName) %>"
    workspace="<%=SecurityElement.Escape(strWorkspaceName) %>"
    baseurl="<%=SecurityElement.Escape(baseUrl.AbsoluteUri) %>"
    userdisplayname="<%=GetDisplayName(strDomainUserName) %>"
    >
</RenderFailureMessage>

```

This last modification tells the xsl template that it can expect a new variable, and it’s called “userdisplayname”. The value of that variable is the result of the GetDisplayName code you inserted earlier.

Save and close Default.aspx.

Note that if you want to show the displayname on other pages than Default.aspx you need to make these two changes in those files as well! The modifications above in Default.aspx will only work on Default.aspx.

Now open Site.xsl in an editor, which is in C:\Windows\Web\RDWeb\Pages.

Find line 15 (empty line) and insert the following code:

```
<xsl:variable name="userdisplayname" select="/RDWAPage/@userdisplayname"/>
```

The result will look like this:

```
<xsl:variable name="appfeedcontents" select="/RDWAPage/AppFeed[1]"/>
<xsl:variable name="strings" select="document(concat($baseurl, 'RDWAStrings.xml'))/str:strings/string"/>
<xsl:variable name="userdisplayname" select="/RDWAPage/@userdisplayname"/>

<!-- top level document structure -->
```

Here we grab the new variable which we defined in Default.aspx (and now you see why we need to do this on every page that we want to show the displayname on, because the variable is supplied by the actual page).

Find line 323:

```
<xsl:value-of select="$strings[@id = 'SignOut']"/>
```

Add the following code on the end of that line:

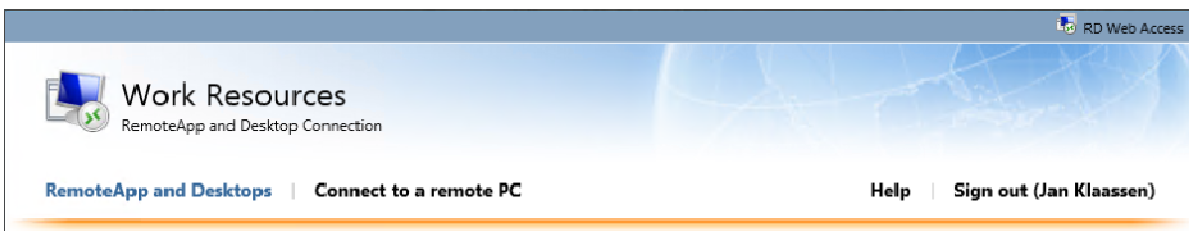
```
<xsl:if test="$userdisplayname"> (<script type="text/javascript"
language="javascript">document.write(decodeURIComponent('<xsl:value-of
select="$userdisplayname"/>'));</script></xsl:if>
```

The result must look like this:

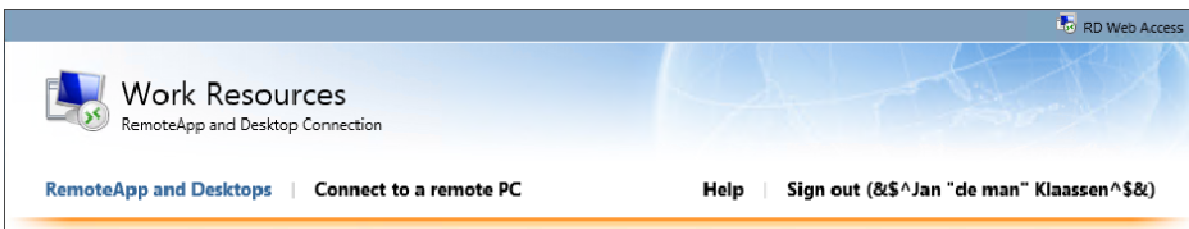
```
<a id='PORTAL SIGNOUT' href="javascript:onUserDisconnect()" target="_self">
  <xsl:value-of select="$strings[@id = 'SignOut']"/><xsl:if test="$userdisplayname"> (<script type=
  "text/javascript" language="javascript">document.write(decodeURIComponent('<xsl:value-of
  select="$userdisplayname"/>'));</script></xsl:if>
</a>
```

Save and close Site.xsl.

Open your RD Web Access page, login using valid credentials:



Or, with special characters:



Nice! We now have the displayname displayed on the navigation bar, right next to the Sign out button.

I have tested this in my very simple lab setup. I have only one domain, with a single UPN context, so I couldn't test this with aliases and such.

I have tested this customization using DOMAIN\username, UPN and just username, and that all works.