CF8 and LDAP snippets

Posted At : May 24, 2011 10:24 AM | Posted By : Sid Wing
Related Categories: Microsoft, Coldfusion, Programming

Working in a predominately Windows-based network can sometimes make life a bit challenging...

Anyone who has ever worked with Web Apps that need to integrate with Windows Active Directory, I am sure, has run into some level of frustration with the way Microsoft has chosen to store certain fields. First, let's take a look at 3 "interesting" little Active Directory fields.

The first AD (Active Directory) field I want to look at is "LastLogon". This field contains the date (sort of) that a User last logged into the network (Really??? I would never have know that from the field name). So, let's query our AD first:

```
<cfldap server = "192.168.1.1"
port="389"
action = "query"
name = "results"
start = "dc=mydomain,dc=net"
scope="subtree"
returnasbinary="objectGUID objectSID"
filter="(&(objectClass=user)(!(!(givenName=*))))"
attributes = "accountExpires, description, whenChanged, cn, mail, sn, givenName,
telephonenumber, whenCreated, physicalDeliveryOfficeName, SamAccountname, pwdLastSet, dn,
UserAccountControl, lastLogon, badPasswordTime, objectGUID, objectSID"
sort = "cn ASC"
username="myadmin"
password="myPassw0rd!">
```

This query will return a NUMBER of variables on all "USER objects" in AD that have a "givenName" (first name) entry (this thins out some of the accounts in AD that are system or network accounts). One of the variables it returns is "lastLogon". In its "natural state" - this comes back as a number that is resembles something like this: 129461445741836159. Doesn't look much like a date, does it?

Well - it's NOT a date - it's a TIMESPAN. It is the "number of 100 nanosecond intervals since 12:00 AM January 1, 1601" to the last time the user logged on to the network. Now THAT'S what I call a "friendly number" to work with... {sarcasm}

So... In order to turn this into something that can actually be USED by a HUMAN - we have to "massage" this number into a REAL date. After a couple of days research online and MUCH frustration - I finally found an article that pointed me in the right direction - but it was written in VBScript. So I set about converting it into the code displayed below:

```
<cffunction name="convertNumberToDate" access="public" returntype="date" output="false"
hint="Convert Last Logon/PwdReset to a date">
<cfargument name="ThisValue" required="true" />
<!--- This is an arbitrary date chosen as the "base" of date calculation. --->
<cfset DatumDate = CreateDate(2000, 1, 1)>
<!--- This values equates to the number of seconds from when Windows begins --->
<!--- time calculations. This actual value equates to the date of 1/1/2000. --->
<cfset DatumOffset = 12591266400>
<cfif IsNumeric(ThisValue) and Len(ThisValue) eq 18>
    <!--- Pull the first 11 digits of the time value. This gets us down to seconds. --->
    <cfset Offset = Left(ThisValue, 11)>
    <!--- Get the difference between the datum value and the current value. --->
    <!--- Add 86400 to add another day to the value for the correct result. --->
    <cfset DiffSeconds = (Offset - DatumOffset) + 86400>
<cfelse>
    <cfif ThisValue is not ''>
    <cfset DiffSeconds = ThisValue>
    <cfelse>
    <cfset DiffSeconds = 0>
    </cfif>
</cfif>
<cfset myRealDate = #DateFormat(DateAdd("s", DiffSeconds, DatumDate))#>
<!--- End LastLogon Convert --->
<cfreturn myRealDate />
</cffunction>
```

This function requires only 1 argument: (ThisValue). (ThisValue) is the ugly, long, number stored (in this case at least) in the "lastLogon" AD Field. The function creates a date called DatumDate based on a known date (1/1/2000). Then it sets DatumOffset to yet another large number - 12591266400 - the number of seconds between 1/1/1601 and 1/1/2000. Next, we check that (ThisValue) is a numeric value and that it has a length of 18 (I will explain why the length is important in my next post on this subject). Once that is done - we get the first 11 characters (Offset) of (ThisValue), which equates itself to the number of seconds between the user's last logon and midnight on 1/1/1601.

Now we can set (DiffSeconds) to be (Offset {the number of seconds from 1/1/1601 to last logon}) - (DatumOffset {the known value of seconds between midnight 1/1/1601 to midnight 1/1/2000}) (so in our example, this would be (12946144574 - 12591266400) + 86400). The "+86400" adds another day's worth of seconds to the equation to compensate for some bizarre calendar math that occured with leap years and leap centuries. So that leaves us with a total of 354964574 seconds stored in (DiffSeconds).

Now we can create a real date (myRealDate) by adding the remaining seconds (Diffseconds) to the known date (DatumDate) - and then using the DateFormat function to convert that to the standard locale date format.

Once ALL of this is complete, we can return the REAL DATE to the page that calls this function.

This function, also, works for converting the value of the "pwdLastSet" AD field in the same manner. In my next post, I will show the additional coding required to handle the "accountExpires" AD field.