Active Directory – Single Sign on using 4D 2004

By Thomas Maul, General Manager, 4D Germany.

TN 06-02

Summary

MS Windows ドメイン環境では、コンピュータで認証を済ませた後はパスワードの入力を求められることなく、ファイルサーバ、プリンタ、メールサーバ、データベースなどのネットワークリソースを自由に利用できるシングルサインインを実施することができます。

アクティブディレクトリにおけるユーザの削除やグループの変更は、ネットワーク全体に影響が及ぶので、管理者は、ユーザ情報を一括管理することができます。

このテクニカルノートでは、プラグインを使用することなく、4D アプリケーションを Active Directory と連携させる方法について論じています。

Benefit for your customers

Microsoft 社のドキュメント"中小企業で Active Directory を導入する利点について"には次のように記載されています。

Active Directory は、Microsoft Windows Server 2003 および Microsoft Windows 2000 Server に同梱されている統合かつ分散されたディレクトリ サービスです。以前は、管理するために別の独立したディレクトリやユーザー ID/パスワードを必要としていた多くのアプリケーションやサービスが Active Directory に統合されました。たとえば、Windows NT 4.0 では、ドメイン自体にディレクトリが必要で、Exchange メールボックスや配布リスト用に個別のディレクトリが必要で、リモート アクセス、データベース、他のアプリケーション用にも個別のディレクトリが必要でした。また、アプリケーションごとに、別のパスワードが必要な場合もありました。Active Directory を使用すると、組織の管理者は、Active Directory にユーザーを追加し、その 1 つのエントリを使用してネットワークへのリモート アクセスを有効にしたり、Exchange メッセージングで同じユーザー アカウントを有効にしたり、その同じアカウントを使用して、会計、クライアント関係管理、または他の用途のアプリケーションで使用するデータベースへのアクセスを提供することができます。Active Directory は、このように多目的のディレクトリとして使用することができるだけでなく、組織では、Active Directory を多目的のディレクトリとして使用することでユーザーにシングル サインオンの機能を提供することができます。ユーザーが Windows にログインすると、ユーザーの Active Directory の資格情報に基づいて、アプリケーションやサービスのロックが解除されます。これには、Windows 統合認証を使用するサード パーティ アプリケーションも含まれます。

出典:http://www.microsoft.com/japan/windowsserver2003/techinfo/overview/adsmallbiz.mspx

加えて、アプリケーションのセキュリティ向上も挙げることができます。ある種のカスタムソリューションでは、一般的なユーザ ID/パスワード以上のセキュリティが求められます。スマートカードや指紋による認証システムは、加速度的に普及しつつあります。Windows Active Directory でサポートされているアクセスシステムは、追加作業を要することなく、どのアプリケーションからもサポートされます。

Active Directory

MS Windows コンピュータは、ローカルユーザリストを持つスタンドアロンマシン、あるいは中央で管理されたグループポリシーとドメインユーザリストを共有するドメインメンバーのどちらかになります。

サンプルコードを実行するには、ドメインコントローラ(通常は Windows 2000、2003 Server、または Small Business Server)が必要で、使用する Windows マシンはそのドメインのメンバーでなければなりません。スタンドアロンの Windows コンピュータの場合、コードを実行することはできますが、当然のこととしてドメインコントローラからグループメンバーシップを取得することはできません。

Important: Mac OS クライアントには対応していません。

Usage with 4D – Method AD_UserGroups

このテクニカルノートで紹介しているメソッドは、下記の情報を返します。

- カレントの認証されたユーザ名
- カレントのドメイン名
- カレントのドメイン DNS サフィックス
- ユーザのグループリスト

```
$out:=""
C_BLOB($result)
PLATFORM PROPERTIES($plat)
If ($plat#3)
   ALERT("This code uses Active Directory. It can be used on Windows only")
   Else
   $path:=Temporary folder+"myscript.vbs"
   $status:=Test path name($path)
   If ($status=1)
      DELETE DOCUMENT($path)
   End if
   $ref:=Create document($path)
   $cr:=Char(13)+Char(10)
```

```
    $text:="Dim WshNetwork"+$cr
    $text:=$text+"Dim FSO"+$cr
    $text:=$text+"Dim strUserName"+$cr
    $text:=$text+"Dim strUserDomain"+$cr
    $text:=$text+"Dim strDNSDomain"+$cr
    $text:=$text+"Dim bjRootDSE"+$cr
    $text:=$text+"Set WshNetwork = WScript.CreateObject(¥"WScript.Network¥")"+$cr
    $text:=$text+"Set FSO = CreateObject(¥"Scripting.FileSystemObject¥")"+$cr
    $text:=$text+"strUserName = WSHNetwork.UserName"+$cr
    $text:=$text+"While strUserName = ¥"¥""+$cr
    $text:=$text+"WScript.Sleep 100"+$cr
    $text:=$text+"strUserName = WSHNetwork.UserName"+$cr
    $text:=$text+"Wend"+$cr
    $text:=$text+"strUserDomain = WSHNetwork.UserDomain"+$cr
    $text:=$text+"WScript.StdOut.Write strUserName & vbCR & vbLF"+$cr
    $text:=$text+"WScript.StdOut.Write strUserDomain & vbCR & vbLF"+$cr
    $text:=$text+"Set objRootDSE = GetObject(¥"LDAP://RootDSE¥") "+$cr
    $text:=$text+"strDNSDomain = objRootDSE.Get(¥"defaultNamingContext¥")"+$cr
    $text:=$text+"WScript.StdOut.Write strDNSDomain & vbCR & vbLF"+$cr
    $text:=$text+"Set CreateMemberOfObject = CreateObject(¥"Scripting.Dictionary¥")"+$cr
    $text:=$text+"CreateMemberOfObject.CompareMode = vbTextCompare"+$cr
    $text:=$text+"Set objUser = GetObject(¥"WinNT://¥" & strUserDomain & ¥"/¥" & strUserName &
¥",user¥")"+$cr
    $text:=$text+"For Each objGroup In objUser.Groups"+$cr
    $text:=$text+"WScript.StdOut.Write objGroup.Name & vbCR & vbLF"+$cr
    $text:=$text+"Next"+$cr
    SEND PACKET($ref;$text)
    CLOSE DOCUMENT($ref)
    $out:=""
    $input:=""
    $err:=""
    $cmd:="cscript //Nologo "+Char(34)+$path+Char(34)
    SET ENVIRONMENT VARIABLE("_4D_OPTION_HIDE_CONSOLE";"true")
    LAUNCH EXTERNAL PROCESS($cmd;$input;$out;$err)
    DELETE DOCUMENT($path)
    $out:=Replace string($out;Char(13)+Char(10);Char(13))
    $out:=Win to Mac($out)
    If ($err#"")
       $out:="Error: "+$err+Char(13)+Char(13)+$out
    End if
    If (Count parameters>0)
       Case of
       ¥ (Type($1->)=Text array )
          ARRAY TEXT($1->;0)
          $pos:=Position(Char(13);$out)
          While ($pos>0)
             APPEND TO ARRAY($1->;Substring($out;1;$pos-1))
             $out:=Substring($out;$pos+1)
             $pos:=Position(Char(13);$out)
          End while
          If ($out#"")
             APPEND TO ARRAY($1->;$out)
          End if
       ¥ (Type($1->)=ls Text )
          $1->:=$out
       End case
    End if
 End if
```

データはテキストまたはテキスト配列として返されます。

Example:

Thomas
DE4D
DC=de,DC=4D,DC=local
SBS Mobile Users
4DGmbH
Web Workplace Users
Domain Admins
Domain Power Users
Domain Users
Designer

- 第 1 行/要素は Windows ログイン画面に入力され、Active Directory の admin で作成されるユーザ名です。
- 第 2 行/要素はユーザがサインインしているドメインの名前です。ユーザがドメインのメンバーでなければ、コンピュータの名前が返されます。
- 第 3 行/要素は LDAP 通信で使用されるシンタックスによるドメインの DNS サフィックスです。上記の例の場合、サフィックスは"de.4D.local”です。ユーザがドメインのメンバーでなければ、この行/要素にはエラーメッセージが返され、返り値の最終行/要素となります。
- 残りの行/要素はユーザが所属しているすべてのセキュリティグループの名前が返されます。配布グループなど、その他のグループの名前は返されません。

メソッドは次のようにコールできます。

ARRAY TEXT(arrText;0)

AD_UserGroups (->arrText)

または

C_TEXT(vText)

AD_UserGroups (->vText)

メソッドにはテキスト変数またはテキスト配列に対するポインタを渡します。

Using your own password system

On Startup でカスタムダイアログを表示し、ユーザ名とパスワードの入力を求めるようなデータベースがあるとしましょう。この場合、ダイアログを表示する前に *AD_UserGroups* メソッドをコールし、システムで許可されているユーザであればダイアログを表示しないでログインをするようにコードを変更します。許可されていないユーザであれば、顧客のセキュリティガイドラインに従い、ログインを拒否、または特殊なダイアログを表示するようにします。

次のようなコードを記述することができるでしょう。

```
$domain:=[Prefs]Domainname
$domainsuffix:=[Prefs]Domain_Suffix
$groupname:=[Prefs]Main_Group
PLATFORM PROPERTIES($plat)
If (($plat=3) & (Not(Shift down)))
   ` this gives us a chance to use the system on Mac
   ` or Windows computers not belonging to a domain server
   ARRAY TEXT($users;0)
   AD_UserGroups (->$users)
      If (Size of array($users)>3) ` else cannot be a legal login
         If (($users{2}=$domain) & ($users{3}=$domainsuffix))
         $pos:=Find in array($users;$groupname;4)
            If ($pos>0)
            ` User exist and is in the group - so let him in
```

この行まで到達したのであれば、ユーザはデータベースの使用を認められたユーザです。無論、データベースを利用できる程度には差を設ける必要があるかもしれません。Else 部分には必要に応じて様々な処理を記述することができます。そこでは、ダイアログを表示したり、QUIT 4D をコールしたりします。

その後、メソッドは以下のように処理を続けます。

```
            ` User exist and is in the group - so let him in
            $helpblob:=[Prefs]Adminpassword
            DECRYPT BLOB($helpblob;[Prefs]Private_key;[Prefs]Public_key)
            $password:=BLOB to text($helpblob;Text without length )
            CHANGE CURRENT USER([Prefs]Adminuser;$password) ` this gives us the right to
create/modify users
            `check the current valid/assigned groups
            ARRAY TEXT($groupnames;0)
            ARRAY LONGINT($grouplist;0)
            ARRAY LONGINT($grouplistold;0)
            GET GROUP LIST($groupnames;$grouplist)
            For ($i;1;Size of array($groupnames))
               $pos:=Find in array($users;$groupnames{$i};4)
               If ($pos<1)
                  $groupnames{$i}:=""
               End if
            End for
            $pos:=Find in array($groupnames;"")
            While ($pos>0)
               DELETE ELEMENT($groupnames;$pos)
               DELETE ELEMENT($grouplist;$pos)
               $pos:=Find in array($groupnames;"")
            End while
            `check if user exists in 4D password system, else create
            $Username:=$users{2}+"/"+$users{1}
            ARRAY TEXT($Userlist;0)
            ARRAY LONGINT($UserID;0)
            GET USER LIST($Userlist;$userID)
            For ($i;1;Size of array($userID))
               If (Is user deleted($UserID{$i}))
                  $Userlist{$i}:="xxxxxxxxxxxxx"
               End if
            End for
            $pos:=Find in array($Userlist;$Username)
            If ($pos<0)
               $password:=""
```

```
                For ($i;1;15)
                   $password:=$password+Char((Random%(90-65+1))+65)
                End for
                $newID:=Set user properties(-2;$Username;"";$password;0;!00/00/00!;$grouplist)
                QUERY([Users];[Users]Name=$Username)
                If (Records in selection([Users])=0)
                   CREATE RECORD([Users])
                   [Users]Name:=$Username
                   [Users]ID:=$newID
                End if
                TEXT TO BLOB($password;[Users]Password;Text without length )
                ENCRYPT BLOB([Users]Password;[Prefs]Private_key;[Prefs]Public_key)
                SAVE RECORD([Users])
             Else
                $newID:=$userID{$pos}
                QUERY([Users];[Users]Name=$Username)
                DECRYPT BLOB([Users]Password;[Prefs]Private_key;[Prefs]Public_key)
                $password:=BLOB to text([Users]Password;Text without length )
                ` check if groups are still valid
                GET USER PROPERTIES($newID;$name;$startup;$passdummy;$login;$logdate;$group
listold)
                $modified:=False
                If (Size of array($grouplist)#Size of array($grouplistold))
                   $modified:=True
                Else
                   For ($i;1;Size of array($grouplistold))
                      If ($grouplistold{$i}#$grouplist{$i})
                         $modified:=True
                      End if
                   End for
                End if
                If ($modified)
                   $newID:=Set user properties($newID;$name;"";$password;$login;$logdate;$groupli
st)
                End if
             End if
             CHANGE CURRENT USER("__login";"")
             ` login back, just to be sure that if login fails, we have no rights
             CHANGE CURRENT USER($Username;$password)
          End if
       End if
    End if
End if
If (($newID=0) & (Current user="__login")) ` this is only the case on Mac or without domain
    CHANGE CURRENT USER
End if
If (Current user="__Login")
    ALERT("Sorry, there was a problem with login, please try again…")
    QUIT 4D
Else
    ` do your startup job here…
    ` Startup
End if
```

Using 4D's password system

4D のパスワードシステムを使用することには、少なくとも 3 つの利点があります。同システムを使用すれば、トリガを含め、コード中のどこからでも Current user 関数をコールしてカレントユーザを知ることができ、ログファイルを調べればレコードを更新したユーザを知ることが

でき、LOCKED ATTRIBUTES コマンドでレコードをロックしているユーザを調べることができます。
4D のパスワードシステムといっても、4D のパスワードエディタ(ツールボックス)やログインダイアログを使用する必要はありません。4D 2004 で変更された CHANGE CURRENT USER コマンドでは、4D のシステムを完全にユーザの目から隠すことができます。

Active Directory と連携する場合、次のようにして 4D のパスワードシステムを隠すことができます。つまり、新しいユーザを自動的に作成し、グループに割り当てた後に、グループから取り除いて内部的にログインを実行します。

このようにすれば、Active Directory Administrator は自由にユーザを作成(削除)したり、責任に応じてグループに割り当てることができ、4D のパスワードエディタやグループエディタを使用しなくても、設定が自動的にデータベースで継承されます。

こうした仕組みにはある程度のコーディングが求められます。サンプルデータベースのメソッド *Startup_Example* はそうしたソリューションの一例です。導入にあたっては、開発者だけが知っているパスワードを Designer に設定します。Administrator の定義済パスワードは、コードの中でのみ使用され、エンドユーザの目に触れることはありません。Administrator ユーザアカウントを本物のユーザが使用することも決してありません。第 3 の定義済ユーザは、パスワードが未設定のデフォルトユーザなので、4D はログインダイアログを省略して On Startup を実行します。このユーザには何のアクセス権限もなく、どのグループにも属していません。このユーザは On Startup を実行するためだけに存在します。

–ログインの手順は次とおりです。
- ユーザが許可されたドメインのドメインメンバーであることを確認する
- ユーザがデータベースを使用できるグループに所属していることを確認する
- CHANGE CURRENT USER で 4D の Administrator としてログインする
- 4D のグループ情報を読み取る–ユーザが所属するグループ名の配列を作成する
- ユーザが 4D のパスワードシステムに存在するかを調べ、存在しなければ作成し、ランダムに作成されたパスワードを暗号化してレコード保存する
- ユーザが所属するグループを調べ、該当する 4D のグループを割り当てる
- CHANGE CURRENT USER で対応するユーザとしてログインする
- この時点で Current user を調べ、デフォルトユーザであれば不正なアクセスとみなし、QUIT 4D を実行する

About the method AD_UserGroups

メソッドではネットワーク情報を読み取るために VBS スクリプトを使用しています。スクリプトは Windows Network オブジェクトを作成し、オブジェクトからカレントユーザとドメイン名を取得します。次いでこの情報を利用して、ネットワークからグループメンバーシップを、そして最終的には LDAP でルートドメインの DNS サフィックスをリクエストします。
結果はテキストで 4D に渡され、テキスト変数にそのまま代入されて返されるか、要素に分割されて配列として返されます。