



Customizing DMA with Scripting

White Paper

May 22, 2001

Contents

Scripting Objects.....	1
Scripting Events	1
Using Microsoft ADSI	2
Using the Settings Object	5

The NetIQ Domain Migration Administrator product (Domain Migration Administrator) provides a flexible environment in which you can customize the migration process to meet your specific needs. You can write customized processing scripts that Domain Migration Administrator can then run when specific events occur during the migration process:

Pre-migration events

Allow a script to check each user account, group, or computer account and determine whether it should be migrated.

Post-migration events

Allow a script to set additional custom properties for an object after that object has been migrated to a Windows NT or Windows 2000 domain.

Domain Migration Administrator supports both VBScript and JavaScript so you can leverage your existing scripting expertise. For more information about VBScript or JavaScript, see the resources available throughout the market, such as on the Microsoft Web site (www.microsoft.com).

Legal Notice

THIS DOCUMENT AND THE SOFTWARE DESCRIBED IN THIS DOCUMENT ARE FURNISHED UNDER AND ARE SUBJECT TO THE TERMS OF A LICENSE AGREEMENT OR A NON-DISCLOSURE AGREEMENT. EXCEPT AS EXPRESSLY SET FORTH IN SUCH LICENSE AGREEMENT OR NON-DISCLOSURE AGREEMENT, NETIQ CORPORATION PROVIDES THIS DOCUMENT AND THE SOFTWARE DESCRIBED IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW DISCLAIMERS OF EXPRESS OR IMPLIED WARRANTIES IN CERTAIN TRANSACTIONS; THEREFORE, THIS STATEMENT MAY NOT APPLY TO YOU.

This document and the software described in this document may not be lent, sold, or given away without the prior written permission of NetIQ Corporation, except as otherwise permitted by law. Except as expressly set forth in such license agreement or non-disclosure agreement, no part of this document or the software described in this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, or otherwise, without the prior written consent of NetIQ Corporation. Some companies, names, and data in this document are used for illustration purposes and may not represent real companies, individuals, or data.

This document could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein. These changes may be incorporated in new editions of this document. NetIQ Corporation may make improvements in or changes to the software described in this document at any time.

© 1995-2001 NetIQ Corporation, all rights reserved.

U.S. Government Restricted Rights: The software and the documentation are commercial computer software and documentation developed at private expense. Use, duplication, or disclosure by the Government is subject to the terms of the NetIQ standard commercial license for the software, and where applicable, the restrictions set forth in the Rights in Technical Data and Computer Software clauses and any successor rules or regulations.

ActiveAgent, ActiveEngine, ActiveKnowledge, ADcheck, AD Modeler, AppAnalyzer, Application Scanner, AppManager, the AppManager logo, Chariot, Directory and Resource Administrator, Domain Migration Administrator, End2End, Enterprise Administrator, Event Manager, Exchange Administrator, Exchange Migrator, File Administrator, File and Storage Administrator, Ganymede, Ganymede Software, the Ganymede logo, Knowledge Pack, Knowledge Scripts, Migration Assessor, Mission Critical Software, the Mission Critical Software logo, MP3check, NetIQ, the NetIQ logo, NetIQ Partner Network, the NetIQ Partner Network logo, NetWare Migrator, OnePoint, the OnePoint logo, Operations Manager, OU Populator, Pegasus, Qcheck, Security Manager, Server Consolidator, and Work Smarter are trademarks or registered trademarks of NetIQ Corporation or its subsidiaries in the United States and other jurisdictions. All other company and product names mentioned are used only for identification purposes and may be trademarks or registered trademarks of their respective companies.

Scripting Objects

Domain Migration Administrator supports the following objects for scripting:

- SourceObject** A Microsoft ADSI IADs pointer to the user account, group, or computer account defined in the source domain. This object is available during pre- and post-migration events.
- TargetObject** A Microsoft ADSI IADs pointer to the user account, group, or computer account defined in the target domain. This object is available only during post-migration events. During pre-migration events, the target object has not yet been created.
- Settings** A NetIQ object containing the setting used for the migration. This object is available during pre- and post-migration events. Scripts can read values from the settings object, but they generally should not change values in the settings object. For more information, see “Using the Settings Object” on page 5.

Note:

Domain Migration Administrator does not run scripts when you select **Test the migration settings and migrate later** on the Test or Make Changes window.

Scripting Events

Domain Migration Administrator supports six specific scripting events:

- Pre-migration of a user account
- Pre-migration of a group
- Pre-migration of a computer account
- Post-migration of a user account
- Post-migration of a group
- Post-migration of a computer account

Pre-Migration Events

Pre-migration events trigger a script to run before the object is migrated. Use pre-migration scripts can make changes to the source object before it is migrated. Each pre-migration script is implemented as a **PreProcess** subroutine. When you first select each pre-migration event, Domain Migration Administrator shows an empty **PreProcess** subroutine, which you can replace with your script.

Post-Migration Events

Post-migration events trigger a script to run after Domain Migration Administrator creates the target object and copies the properties, but before Domain Migration Administrator updates the group memberships. Post-migration events can modify either the source account or the target account. Each event is implemented as a **Process** subroutine. When you first select each post-migration event, Domain Migration Administrator shows an empty **Process** subroutine, which you can replace with your script.

Note:

If you rename or move the target object, you must update the Domain Migration Administrator settings object with the new object path. Otherwise, Domain Migration Administrator will not have the correct path for the object, and group memberships will not be updated for that object. For more information, see "Using the Settings Object" on page 5.

Using Microsoft ADSI

You can use industry-standard scripting technologies, such as VBScript and Microsoft ADSI, to perform most Domain Migration Administrator scripting tasks. For more information about using ADSI, see the ADSI documentation available on the Microsoft Web site at the following URL:

<http://www.microsoft.com/windows2000/techinfo/howitworks/activedirectory/adsilinks.asp>

Domain Migration Administrator uses a specific provider based on the operating system.

Operating system	Provider used by Domain Migration Administrator
Windows NT 4.0	Microsoft WinNT ADSI provider
Windows 2000	Microsoft Active Directory LDAP provider

The following sections provide several example scripts using VBScript and ADSI.

Changing the SAMAccountName and UPN Name to Lower Case

This script is implemented as a post-migration event, since it modifies the target object.

```
' This script changes the samAccountName and the UPNs to lower case.
Sub Process
' Get the current samAccountName field from the target object
  sam = TargetObject.get("samAccountName")

  ' Make the samAccountName all lower case
  newSam = LCase(sam)

  ' Update the target object with the new value
  TargetObject.put "samAccountName", newSam

  ' Same thing for the UPN
  upn = TargetObject.get("userPrincipalName")
  newUpn = LCase(upn)
  TargetObject.put "userPrincipalName", newUpn

  ' Save the changes to AD
  TargetObject.SetInfo
End Sub
```

Prepend a Text String to the Full Name of the Source User Account after Migration

This post-migration script assumes that the source domain is running Windows NT 4.0. For Windows 2000 source domains, you should change the **displayName** property instead of the **FullName** property.

```
Sub Process
  Dim fullname
  Dim prefix

  ' Define a prefix to be added to the full name.
  prefix = "*** Migrated ***"

  ' Get the full name of the source account.
  fullname = sourceObject.Get("FullName")

  ' Add the prefix.
  fullname = prefix & fullname

  ' Update the source object with the new information.
  sourceObject.Put "FullName", fullname

  ' Save the changes.
  sourceObject.SetInfo
End Sub
```

Changing the Logon Script Field Based on Global Group Membership

This post-migration script changes the logon script field to a new value, determined by the membership of a source global group.

```
' Edit aGlobalGroups and aScriptPaths to indicate the groups
' you would like to search for membership in and the scripts
' you would like to be associated with the members of these groups.

' To set a default script path for users that are not a member
' of any of the specified groups, edit the sDefaultPath parameter.
' If the sDefaultPath parameter is left blank,
' the scriptPath will not be changed.
```

Sub Process

```
Dim sNewScriptPath
Dim oGroup
Dim bContinue
Dim i
Dim nArraySize
Dim sDefaultPath
Dim aGlobalGroups
Dim aScriptPaths
Dim sSourcePath

' The aGlobalGroups array must be configured with the paths of the
' global groups in which to check membership of the migrated user.
' The aScriptPaths array should contain the name of the script that
' should be associated with the group in the aGlobalGroups array
' that has the same index.

aGlobalGroups = Array("WinNT://DOMAIN/Group1",
    "WinNT://DOMAIN/Group2")
aScriptPaths = Array("ScriptPath1","ScriptPath2")

'If the migrated user is not a member of one of the groups specified
'scriptPath should be set to the following default value
sDefaultPath = ""

'Initialize variables
sNewScriptPath= ""
sSourcePath = SourceObject.ADsPath
bContinue = true
i = 0
nArraySize = UBound(aGlobalGroups)
'Make sure each global group has a script path to match it
If (UBound(aGlobalGroups) <> UBound(aScriptPaths)) Then
    MsgBox "There is not a one-to-one mapping between global groups
        and script paths"
    Exit Sub
End If
```

```

'Loop through the groups in the array and determine
'If the object is a member of any of the groups
Do While (bContinue And (i <= nArraySize))
    Set oGroup = GetObject(aGlobalGroups(i))
    If (oGroup.IsMember(sSourcePath)) Then
        'Found a group that the object belongs to
        sNewScriptPath = aScriptPaths(i)
        bContinue = false
    End If
    i = i + 1
Loop
'Set the scriptPath on the migrated object
If bContinue AND (sDefaultPath <> "") Then
    'The migrated user is not a member of any of the groups
    TargetObject.Put "scriptPath", sDefaultPath
Else
    If sNewScriptPath <> "" Then
        'The migrated user is a member of one of the groups
        TargetObject.Put "scriptPath", sNewScriptPath
    End If
End If

'Commit the changes to the directory
TargetObject.SetInfo
End Sub

```

Using the Settings Object

The settings that define the migration task are stored in the settings object. You can store and retrieve named values from the settings object. Most Domain Migration Administrator scripts do not need to use the settings object. There are several reasons you may need to use the settings object:

- You want to make your scripts more flexible by pulling dynamic values from Domain Migration Administrator rather than using hardcoded names.
- If your script moves or renames the target object, the script must update the Domain Migration Administrator settings with the new object name and path. If Domain Migration Administrator does not have the new object name and path, Domain Migration Administrator cannot correctly update the group memberships for that object.
- You want to exclude an object from the migration.

To set values in the settings object, use the **put** method. To retrieve values from the settings object, use the **get** method. In most cases, scripts should not modify values in the settings object, as these modifications can adversely affect the migration. To see a list of values that are defined in the settings object, perform a migration task and then view in the settings table in the **protar.mdb** database.

The following sections provide several example scripts that use the settings object.

Retrieving the Source Domain from the Settings Object

This script is implemented as a post-migration event, but it could also be implemented as a pre-migration event.

```
Sub Process
    srcdom = Settings.get("Options.SourceDomain")
    MsgBox("The source domain is " & srcdom)
End Sub
```

Prepend a Text String to the Full Name of the Source User Account after Migration

This post-migration script is similar to the script example in “Prepend a Text String to the Full Name of the Source User Account after Migration” on page 3. However, this script includes the domain name in the prefix string.

```
Sub Process
    Dim fullname
    Dim tgtdom
    Dim prefix

    ' Define a prefix to be added to the full name.
    prefix = "Migrated to "
    tgtdom = settings.Get("Options.TargetDomain")

    ' Get the full name of the source account.
    fullname = sourceObject.Get("FullName")

    ' Add the prefix.
    fullname = prefix & tgtdom & " "& fullname

    ' Update the source object with the new information.
    sourceObject.Put "FullName", fullname

    ' Save the changes.
    sourceObject.SetInfo
End Sub
```

Updating the Object Path in the Settings Object

This post-migration script renames the migrated object so that the common name (**CN**) is set to the full name of the object, rather than the **sAMAccountName** property value.

```
Sub Process
    Dim parentOU
    Dim tempObject
    ' Get an ADSI pointer to the target OU.
    Set parentOU = GetObject(TargetObject.Parent)

    ' Use ADSI to rename the object.
    Set tempObject = parentOU.MoveHere(TargetObject.ADsPath,
    "CN=" & TargetObject.FullName)

    ' Update the settings object with the new name and path.
    Settings.put "CopiedAccount.TargetName",tempObject.Name
    Settings.put "CopiedAccount.TargetPath",
    tempObject.ADsPath

End Sub
```

Excluding an Object from the Migration

This pre-migration script marks objects with the description property set to **test account**, so that Domain Migration Administrator will not migrate those objects.

```
Sub PreProcess
    ' Get the description field of the source object.
    value = sourceObject.Get("Description")

    ' If the description field is set to "test account",
    ' exclude the account from the migration.
    if value = "test account" Then
        Settings.put "CopiedAccount.Operations", 0
    end if
End Sub
```