

# Novell Login Scripts

[www.novell.com](http://www.novell.com)

---

LOGIN SCRIPTS GUIDE

June 2004



**Novell**<sup>®</sup>

## Legal Notices

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

You may not export or re-export this product in violation of any applicable laws or regulations including, without limitation, U.S. export regulations or the laws of the country in which you reside.

Copyright © 1993-2003 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

U.S. Patent No. 5,157,663; 5,349,642; 5,455,932; 5,553,139; 5,553,143; 5,572,528; 5,594,863; 5,608,903; 5,633,931; 5,652,854; 5,671,414; 5,677,851; 5,692,129; 5,701,459; 5,717,912; 5,758,069; 5,758,344; 5,781,724; 5,781,733; 5,784,560; 5,787,439; 5,818,936; 5,828,882; 5,832,274; 5,832,275; 5,832,483; 5,832,487; 5,859,978; 5,870,561; 5,870,739; 5,873,079; 5,878,415; 5,884,304; 5,893,118; 5,903,650; 5,903,720; 5,905,860; 5,910,803; 5,913,025; 5,913,209; 5,915,253; 5,925,108; 5,933,503; 5,933,826; 5,946,002; 5,946,467; 5,956,718; 5,956,745; 5,964,872; 5,974,474; 5,983,223; 5,983,234; 5,987,471; 5,991,810; 6,002,398; 6,014,667; 6,016,499; 6,023,586; 6,029,247; 6,052,724; 6,061,726; 6,061,740; 6,061,743; 6,065,017; 6,081,774; 6,081,814; 6,094,672; 6,098,090; 6,105,062; 6,105,069; 6,105,132; 6,115,039; 6,119,122; 6,144,959; 6,151,688; 6,157,925; 6,167,393; 6,173,289; 6,216,123; 6,219,652; 6,233,859; 6,247,149; 6,269,391; 6,286,010; 6,308,181; 6,314,520; 6,324,670; 6,338,112; 6,345,266; 6,353,898; 6,424,976; 6,466,944; 6,477,583; 6,477,648; 6,484,186; 6,496,865; 6,510,450; 6,516,325; 6,519,610; 6,532,451; 6,532,491; 6,539,381; RE37,178. Patents Pending.

Novell, Inc.  
1800 South Novell Place  
Provo, UT 84606  
U.S.A.

[www.novell.com](http://www.novell.com)

Novell Login Scripts Guide

[December 22, 2003](#)

**Online Documentation:** To access the online documentation for this and other Novell products, and to get updates, see [www.novell.com/documentation](http://www.novell.com/documentation).

## **Novell Trademarks**

AppNotes is a registered trademark of Novell, Inc. in the United States and other countries.

ConsoleOne is a registered trademark of Novell, Inc in the United States and other countries.

eDirectory is a trademark of Novell, Inc.

GroupWise is a registered trademark of Novell, Inc in the United States and other countries.

NetWare is a registered trademark of Novell, Inc. in the United States and other countries.

NetWare Mobile is a trademark of Novell, Inc.

NetWare Requester is a trademark of Novell, Inc.

Novell is a registered trademark of Novell, Inc. in the United States and other countries.

Novell Client is a trademark of Novell, Inc.

Novell Directory Services and NDS are registered trademarks of Novell, Inc. in the United States and other countries.

Virtual Loadable Module and VLM are trademarks of Novell, Inc.

ZENworks is a registered trademark of Novell, Inc in the United States and other countries.

## **Third-Party Trademarks**

All third-party trademarks are the property of their respective owners.



# Contents

- Contents** **5**
- About This Guide** **9**
- 1 Using Login Scripts** **11**
  - Login Script Overview . . . . . 11
  - Where Login Scripts Should Be Located . . . . . 12
  - Common Login Script Commands . . . . . 13
  - Creating or Modifying Login Scripts . . . . . 13
    - Creating or Modifying Login Scripts in iManager . . . . . 14
    - Creating or Modifying Login Scripts in ConsoleOne . . . . . 15
    - Getting Around the One Profile Login Script Restriction . . . . . 16
  - Creating Login Scripts from a Sample or Existing Login Script. . . . . 16
  - Modifying User Login Scripts from the N Menu. . . . . 17
  - Printing Login Scripts . . . . . 18
  - Understanding How Login Scripts Work with NetStorage . . . . . 18
  - Additional Sources for Login Scripts Information . . . . . 19
- 2 Sample Login Scripts** **21**
  - Sample Container Login Script . . . . . 21
  - Sample Profile Login Script. . . . . 22
  - Sample User Login Scripts . . . . . 22
    - Sample User Login Script for Mapping Drives . . . . . 23
  - Sample Default Login Script . . . . . 23
- 3 Login Script Commands and Variables** **25**
  - Login Script Conventions . . . . . 25
  - Using Identifier Variables . . . . . 27
    - Login Parameters with %n Variables . . . . . 30
  - Login Scripts Commands by Task . . . . . 30
    - Network Connection and Resource Access. . . . . 30
    - Login Script Execution . . . . . 31
    - Workstation Environment . . . . . 31
    - Text File Usage . . . . . 31
    - Other . . . . . 32
  - # . . . . . 32
    - Command Format. . . . . 32
    - Example. . . . . 32
  - @ . . . . . 33
    - Command Format. . . . . 33
    - Examples . . . . . 33
  - ATTACH . . . . . 33
  - BREAK . . . . . 34
    - Command Format. . . . . 34
  - CONTEXT . . . . . 34
    - Command Format. . . . . 34

Example . . . . .	34
DISPLAY . . . . .	34
Command Format . . . . .	35
Example . . . . .	35
DRIVE . . . . .	35
Command Format . . . . .	35
Example . . . . .	35
EXIT . . . . .	35
FDISPLAY . . . . .	36
Command Format . . . . .	36
Examples . . . . .	36
FIRE or FIRE PHASERS . . . . .	36
Command Format . . . . .	36
Examples . . . . .	37
GOTO . . . . .	37
Command Format . . . . .	37
Example . . . . .	37
IF...THEN . . . . .	37
Command Format . . . . .	38
Examples . . . . .	39
INCLUDE . . . . .	40
Command Format . . . . .	40
Examples . . . . .	40
LASTLOGINTIME . . . . .	41
Command Format . . . . .	41
MAP . . . . .	41
Command Format . . . . .	42
NO_DEFAULT. . . . .	43
Command Format . . . . .	43
PAUSE . . . . .	43
Command Format . . . . .	43
PROFILE . . . . .	43
Command Format . . . . .	44
Example . . . . .	44
REMARK . . . . .	44
Command Format . . . . .	44
Example . . . . .	44
SCRIPT_SERVER. . . . .	44
Command Format . . . . .	45
SET . . . . .	45
Command Format . . . . .	45
Examples . . . . .	45
SET_TIME. . . . .	46
Command Format . . . . .	46
SHIFT . . . . .	46
Command Format . . . . .	46
Examples . . . . .	46
TERM . . . . .	47
Command Format . . . . .	48
Example . . . . .	48
TREE . . . . .	48
Command Format . . . . .	48
WRITE. . . . .	49
Command Format . . . . .	50
Examples . . . . .	50

<b>4</b>	<b>Troubleshooting Login Scripts</b>	<b>51</b>
	Locating the Problem Statement in the Login Script . . . . .	51
	Turning Off the MAPROOT Function in Windows NT/2000/XP . . . . .	52
	Other Common Login Script Problems . . . . .	52
	Losing Path Statements after Login. . . . .	52
	Cannot Use the LOGIN and NLIST Utilities. . . . .	52
	Cannot Execute External Command . . . . .	52
	LOGIN.EXE Problems . . . . .	52
	EXIT Command Doesn't Work . . . . .	52
	Login Script Does Unexpected Things . . . . .	53





# About This Guide

This guide includes the following information on login scripts that can be used with Novell® eDirectory™ user accounts and Novell Client™ software:

- ♦ Chapter 1, “Using Login Scripts,” on page 11
- ♦ Chapter 2, “Sample Login Scripts,” on page 21
- ♦ Chapter 3, “Login Script Commands and Variables,” on page 25
- ♦ Chapter 4, “Troubleshooting Login Scripts,” on page 51

## Additional Documentation

For documentation on installing and managing Novell Client software, see the *Novell Client for Windows Installation and Administration Guide*.

## Documentation Updates

For the latest version of this documentation, see the [Novell Clients online documentation \(http://www.novell.com/documentation/lg/noclienu/index.html\)](http://www.novell.com/documentation/lg/noclienu/index.html).

## Documentation Conventions

In this documentation, a greater-than symbol (>) is used to separate actions within a step and items within a cross-reference path.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (\*) denotes a third-party trademark.

When a single pathname can be written with a backslash for some platforms or a forward slash for other platforms, the pathname is presented with a backslash. Users of platforms that require a forward slash, such as UNIX\*, should use forward slashes as required by your software.



# 1

## Using Login Scripts

This chapter includes the following information:

- ♦ “Login Script Overview” on page 11
- ♦ “Where Login Scripts Should Be Located” on page 12
- ♦ “Common Login Script Commands” on page 13
- ♦ “Creating or Modifying Login Scripts” on page 13
- ♦ “Creating Login Scripts from a Sample or Existing Login Script” on page 16
- ♦ “Modifying User Login Scripts from the N Menu” on page 17
- ♦ “Printing Login Scripts” on page 18
- ♦ “Understanding How Login Scripts Work with NetStorage” on page 18
- ♦ “Additional Sources for Login Scripts Information” on page 19

### Login Script Overview

A login script is a set of instructions that is executed. A login script is simply a text file that the login executable (login.exe) interprets and runs line by line.

When a user successfully logs in to the network, one or more login scripts are executed which automatically set up the workstation environment. Login scripts are similar to batch files and are executed by the Novell® LOGIN utility.

You can use login scripts to map drives and search drives to directories, display messages, set environment variables, and execute programs or menus.

Login scripts are properties of specific eDirectory™ objects.

There are four types of login scripts:

- ♦ **Container** sets the general environments for all users in that container. Container login scripts are executed first and can be associated with Organization or Organizational Unit objects. A user can use only one container login script.  
**NOTE:** A container login script replaces the system login script from NetWare® 3.
- ♦ **Profile** sets environments for several users at the same time. Profile login scripts are executed after the container login script and are associated with Profile objects. A user can be assigned only one profile login script but can choose other profile login scripts.
- ♦ **User** sets environments (such as printing options or an e-mail username) specific to a single user. User login scripts are executed after any container and profile login scripts and are associated with User objects. A user can have only one user login script.

- ◆ **Default** contains only essential commands, such as drive mappings to NetWare utilities, and cannot be edited. The default login script runs if a user (including user Admin) doesn't have a user login script, even if a container or profile login script exists.

**TIP:** If you don't want to create any user login scripts and you don't want the default login script to execute for any users, you can disable the default login script by including the NO\_DEFAULT command in the container or profile login script.

Maintaining many user login scripts can be time consuming. Therefore, you should try to include as much customization information as possible in the container and profile login scripts, which are fewer in number and easier to maintain.

**IMPORTANT:** Because up to three login scripts can execute whenever a user logs in, conflicts can occur and drive mappings can be overwritten by consecutive login scripts. It is important to note that the last login script to execute (usually the user login script) overrides any conflicting commands in a previous login script.

For example, if all users need access to the NetWare utilities in the same volume, put the search drive mapping to that volume in a single container login script rather than in every user login script.

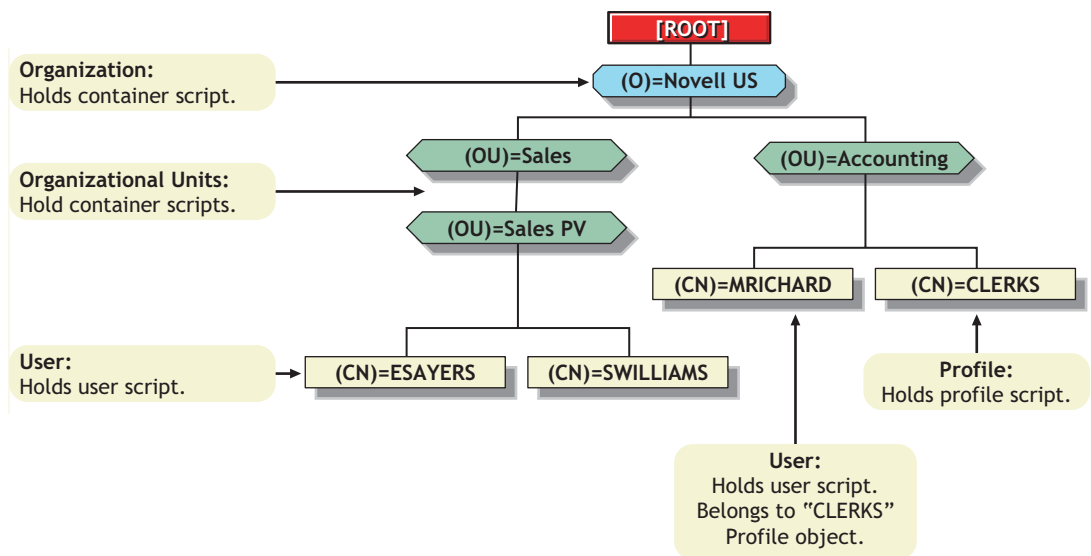
Create profile login scripts if several users have identical login script needs. Profile login scripts are sometimes thought of as group login scripts.

Finally, in user login scripts, include only those individual items that can't be included in profile or container login scripts. For example, personal drive mappings could be included in the user login script.

## Where Login Scripts Should Be Located

Login scripts are properties of objects. Consequently, only certain objects can contain login scripts. This, in turn, largely determines where login scripts can be located.

The following figure shows how the different types of login scripts can reside in an eDirectory tree and how they affect users.



In the figure, there are three users: ESAYERS, SWILLIAMS, and MRICHARD. The following explains which login scripts execute when each of these users logs in:

- ◆ User ESAYERS: The SALES\_PV container login script executes first, followed by ESAYERS's user login script.
- ◆ User SWILLIAMS: The SALES\_PV container login script executes first, followed by the default login script.
- ◆ User MRICHARD: The ACCOUNTING container login script executes, followed by the CLERKS' profile login script, and then MRICHARD's user login script.

Container login scripts affect only users immediately below the Organization or Organizational Unit that contains the login script.

For example, in the previous figure, although there are two levels of container objects above users ESAYERS and SWILLIAMS, only the script for the container they're in (OU=SALES\_PV) executes when they log in.

If the SALES\_PV Organizational Unit had no container login script defined, no container login script would execute for ESAYERS and SWILLIAMS, even though a container login script exists at a higher level.

Because user SWILLIAMS has no user login script defined, the default login script executes after the container login script.

Because user MRICHARD belongs to the profile CLERKS, the CLERKS profile login script executes before MRICHARD's user login script. Users can be assigned to only one Profile object, but there are ways to get around this restriction. See [“Getting Around the One Profile Login Script Restriction” on page 16](#).

## Common Login Script Commands

Login scripts are powerful and can execute many different commands that you might find useful. However, the most common use for login scripts is mapping drives on NetWare servers. The following are the most common commands:

- ◆ **MAP:** Defines a path to a directory on a NetWare server. By putting a MAP command in the login script, the directory is automatically available to the user in My Computer. The drive letter can either be specified (MAP G:=) or can be relative (MAP \*1= or MAP NEXT=). See [“MAP” on page 41](#).
- ◆ **IF...THEN:** Gives statements of condition. For example, IF MEMBER OF “MANAGERS” THEN would execute the command that follows only if the user is a member of the Managers group. See [“IF...THEN” on page 37](#).
- ◆ **INCLUDE:** Provides the name of an ANSI text file to be processed at a specific point in the login script. See [“INCLUDE” on page 40](#).
- ◆ **EXIT:** Terminates all login script processing. EXIT is often used with IF...THEN statements. See [“EXIT” on page 35](#).

Information on these commands and others that you might find useful is provided in [Chapter 3, “Login Script Commands and Variables,” on page 25](#).

## Creating or Modifying Login Scripts

You can use Novell iManager or ConsoleOne<sup>®</sup> to create login scripts from scratch or from a sample login script that is provided for you. Using a sample login script can help reduce syntax errors and thus reduce the time it takes to create login scripts.

All four types of login scripts use the same conventions, commands, and variables. For more information, see [“Login Script Commands and Variables” on page 25](#).

The main difference in creating container, profile, and user login scripts is the object that you select to contain the login scripts. The default login script is not assigned to any object.

- ◆ Container login scripts are assigned to container objects (Organization or Organizational Unit objects).
- ◆ Profile login scripts are assigned to Profile objects. In order for a User object to use a profile login script, you must select that User object and assign it to the Profile object.
- ◆ User login scripts are assigned to User objects.

Before you create or modify login scripts, you must have the Write property right to the object that will contain the login script. In addition, the Organization, Organizational Unit, Profile, or User object that you plan to assign the login script to must already exist.

## Creating or Modifying Login Scripts in iManager

### Creating or Modifying a User Login Script

- 1 Click Roles and Tasks > User Management > Modify User.
- 2 Specify a username and context, then click OK.
- 3 Click General > Login Script.
- 4 Type the login script commands and information in the login script text box.

For a sample, see [“Sample Login Scripts” on page 21](#).

**IMPORTANT:** Make sure that you edit the sample login script to match the server names, directory paths, and specifications of your own network.

For additional information on all login script commands, see [“Login Script Commands and Variables” on page 25](#).

- 5 (Conditional) To associate a Profile login script with this object, specify the Profile object name and context in the Profile field.  
For additional information, see [“Creating a Profile Login Script” on page 15](#).
- 6 To save the login script, click OK.

### Creating a Container Login Script

- 1 Click Roles and Tasks > eDirectory Administration > Modify Object.
- 2 Specify the container and context, then click OK.
- 3 Click General > Login Script.
- 4 Type the login script commands and information in the login script text box.

For a sample, see [“Sample Login Scripts” on page 21](#).

**IMPORTANT:** Make sure that you edit the sample login script to match the server names, directory paths, and specifications of your own network.

For additional information on all login script commands, see [“Login Script Commands and Variables” on page 25](#).

- 5 (Optional) To associate a Profile login script with this object, specify the Profile object name and context in the Profile field.

For additional information, see [“Creating a Profile Login Script” on page 15](#).

- 6 To save the login script, click OK.

## Creating a Profile Login Script

- 1 Click Roles and Tasks > eDirectory Administration > Create Object.
- 2 Select Profile, then click OK.
- 3 Specify the Profile name and context.
- 4 Type the login script commands and information in the login script text box.

For a sample, see [“Sample Login Scripts” on page 21](#).

**IMPORTANT:** Make sure that you edit the sample login script to match the server names, directory paths, and specifications of your own network.

For additional information on all login script commands, see [“Login Script Commands and Variables” on page 25](#).

- 5 Click OK > OK.
- 6 Click Roles and Tasks > User Management > Modify User.
- 7 Specify a username and context, then click OK.
- 8 On the General tab, click Login Script.
- 9 In the Profile field, specify a profile name and context, then click OK.

## Creating or Modifying Login Scripts in ConsoleOne

- 1 Double-click the object whose login script you want to create or modify.  
You can create this login script on a Container object, Profile object, or User object.

- 2 Click Login Script.
- 3 Type the login script commands and information in the login script text box.

For a sample, see [“Sample Login Scripts” on page 21](#).

**IMPORTANT:** Make sure that you edit the sample login script to match the server names, directory paths, and specifications of your own network.

For additional information on all login script commands, see [“Login Script Commands and Variables” on page 25](#).

- 4 To save the login script and close the Details dialog box, click OK.

If the login script that you just created was a container or user login script, you’re finished and the client software will be installed or updated the next time users log in.

If the login script that you just created was for a Profile object, you must associate the User object with the Profile object and make the User object a trustee of the Profile object. See the next section, [“Associating the User Object with a Profile Object.”](#)

For additional information on all login scripts, see [“Creating or Modifying Login Scripts” on page 13](#).

## Associating the User Object with a Profile Object

- 1 Double-click the User object that needs to use the profile login script.
- 2 Click Login Script.
- 3 Type the name of the Profile object in the Default Profile field located under the login script text box, then click OK.
- 4 Add the User object as a trustee of the Profile object by double-clicking the Profile object.
- 5 Click NDS Rights > Trustees of This Object > Add Trustee.
- 6 Specify the name of the User object that uses this Profile object.
- 7 Make sure that the Browse object and the Read Property check boxes are checked, then click OK to assign these rights to the User object.

The User object is now a trustee of the Profile object and has the rights necessary to run the profile login script.

Repeat these steps for all additional users who need to use this script.

## Getting Around the One Profile Login Script Restriction

Users can belong to only one profile login script and so, as a rule, only one profile login script can be executed for any user. There are ways to run the equivalent of different or multiple profile login scripts by specifying other profile login scripts.

For example, to specify a different profile login script for a user at the command line, enter the following:

```
LOGINW32 username /p profile_object
```

You can also assign users to more than one Group object. Use the MEMBER OF group identifier variable to execute different parts of a login script, depending on the Group objects that the user belongs to. To accomplish this, use the If...THEN command (see “[IF...THEN](#)” on page 37) and the MEMBER OF group identifier variable (see “[Using Identifier Variables](#)” on page 27).

To view an example of the IF...THEN command used with the MEMBER OF group identifier variable, see “[Sample Login Scripts](#)” on page 21.

## Creating Login Scripts from a Sample or Existing Login Script

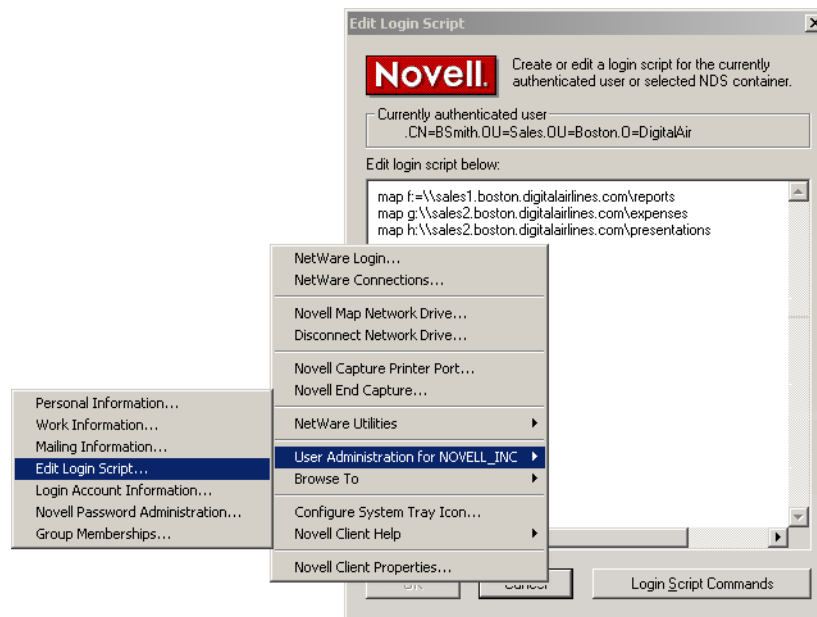
- 1 Locate the sample or existing login script that you want to modify.  
For sample login scripts included in this documentation, see “[Sample Login Scripts](#)” on page 21.
- 2 Copy the sample or existing login script.
- 3 In Novell iManager or ConsoleOne, open the login script, copy the sample, then paste it into the User object or Profile object’s login script.  
See “[Creating or Modifying Login Scripts](#)” on page 13.
- 4 Make necessary changes to the sample login script to customize it for your network.  
**IMPORTANT:** Make sure that you edit the sample login script to match the server names, directory paths, and specifications of your own network.
- 5 Click OK to save the new login script.



If the login script that you just created was a container or user login script, you're finished. If the login script that you just created was for a Profile object, you must associate the User object with the Profile object and make the User object a trustee of the Profile object. See [“Creating a Profile Login Script” on page 15](#) or [“Associating the User Object with a Profile Object” on page 16](#).

## Modifying User Login Scripts from the N Menu

In addition to creating User login scripts in Novell iManager or ConsoleOne, you can also add user login scripts to User objects from the N menu in the system tray. This lets you customize the login script for the user that is currently logged in to the network without running iManager or ConsoleOne.



For example, users can add commands to their login scripts to map drives that they access frequently.

- 1 Log in as the user you want to modify login scripts for.
- 2 Right-click the N menu, then click User Administration for *tree\_name* > Edit Login Script.

- 3 Type the login script commands you want to add.

If you want to map drives, you would type

```
MAP drive_letter:=\\server_name\path_to_volume
```

For example:

```
MAP F := \\MKTG.TOKYO.DIGITALAIRLINE.COM\DOC
```

- 4 Click OK.
- 5 Click Yes to save changes to the login script.
- 6 Log in again to see the changes to your login script.

## Printing Login Scripts

To print a login script from the command line, use the NLIST command and redirect the output to a file or a printer. You must be in an object's parent container to see and print the login script of that object.

**IMPORTANT:** You must have a workstation running DOS 3.30 or later and the Read and File Scan property rights to the object to be printed.

To print a user's login script, enter

```
NLIST user=username show "login script">LPT1
```

To print a container's login script, enter

```
NLIST organizational unit=ou name show login script>LPT1
```

**IMPORTANT:** Any option in the NLIST command that includes a space in its name must be enclosed in quotes.

## Understanding How Login Scripts Work with NetStorage

Novell NetStorage provides secure Internet-based access to files and folders on a NetWare server in your network using either a browser or Microsoft\* Web Folders (Microsoft's implementation of WebDav). NetStorage authentication relies on the power of eDirectory to provide secure access, so Internet-based access is as secure as accessing files from within the network.

When a user authenticates to NetStorage, the login scripts associated with the eDirectory User object in the primary authentication domain are processed. These login scripts are the same scripts processed by the Novell Client.

However, because login scripts were designed to be processed by the Novell Client on the user's workstation, some of the defined statement types and script variables cannot be processed by NetStorage. The following is a list of what login script functions are available when users authenticate using NetStorage.

- ◆ **NetStorage Looks for MAP Commands:** NetStorage processes login scripts in order to find MAP statements. Each MAP statement generates one path into the NetWare file system that the user can access using NetStorage.

In addition to the MAP command, the IF, THEN, ELSE, END, INCLUDE, and EXIT commands are the only commands recognized by NetStorage. All other login script statements are treated as comments and ignored.

- ◆ **Multiple Login Scripts Processed:** When a user successfully logs in to the network, one or more login scripts are executed. Login scripts can be obtained from container objects, Profile objects, User objects, or any file or directory object designated by the INCLUDE statement.
- ◆ **Authentication to Multiple eDirectory Trees:** Users can authenticate to multiple eDirectory trees. One tree is configured as primary. When the user enters a username and password, NetStorage authenticates to the primary tree and, if successful, attempts to authenticate to all other domains.

Only the login scripts from the primary eDirectory tree are processed. However, one of these login scripts might map to a directory on a server that resides in another tree. Because the user is authenticated to all trees, he will also have access to this directory.

- ◆ **eDirectory Variables:** The value of an eDirectory attribute of the current User object can be substituted anywhere in the login script. Precede the attribute name with a percent sign (%)

and replace any spaces in the attribute name with an underscore (\_). The following are some attributes that are useful in login scripts:

`%HOME_DIRECTORY`

`%CN`

`%LANGUAGE`

These variables are generally used as part of the MAP statement or the IF statement. For example:

```
IF "%LANGUAGE" = "ENGLISH"
```

```
MAP H:=\\MYSERVER\USERS\%CN
```

The eDirectory syntax types supported by NetStorage for these attributes are path, string, and list. In the case of the list syntax, only the first string in the list is used.

For more specific information on how login scripts are processed by NetStorage, see "[Login Script Processing by NetStorage](http://support.novell.com/cgi-bin/search/searchtid.cgi?/10068983.htm)" TID# 10068983 (<http://support.novell.com/cgi-bin/search/searchtid.cgi?/10068983.htm>).

## Additional Sources for Login Scripts Information

Because login scripts have been used in multiple versions of Novell products over the past 10 years, there is a wealth of information about them available in other documentation. Here are a few sources to check for additional information:

- ♦ *Novell Client for Windows Installation and Administration Guide*
- ♦ "Using NDS User Object Properties in a Login Script" AppNotes® article (<http://developer.novell.com/research/appnotes/1995/may/04/index.htm>)
- ♦ Troubleshooting Login Scripts TID# 1203151 (<http://support.novell.com/cgi-bin/search/searchtid.cgi?/1203151.htm>)
- ♦ Troubleshooting Login Scripts TID# 10012087 (<http://support.novell.com/cgi-bin/search/searchtid.cgi?/10012087.htm>)
- ♦ Login Scripts and Directory Services TID# 2924798 (<http://support.novell.com/cgi-bin/search/searchtid.cgi?/2924798.htm>)
- ♦ Troubleshooting Container Login Scripts TID# 10025702 (<http://support.novell.com/cgi-bin/search/searchtid.cgi?/10025702.htm>)
- ♦ Login Frequently Asked Questions TID# 2907537 (<http://support.novell.com/cgi-bin/search/searchtid.cgi?/2907537.htm>)
- ♦ ATTACH Command TID# 2907537 (<http://support.novell.com/cgi-bin/search/searchtid.cgi?/2910819.htm>)
- ♦ Login Command Not Executing Due to Server Time Issue TID# 2924912 (<http://support.novell.com/cgi-bin/search/searchtid.cgi?/2924912.htm>)

This additional information, in conjunction with this manual, provide you the needed information to construct and troubleshoot login scripts.



# 2

## Sample Login Scripts

You can create login scripts from these sample login scripts using Novell® iManager, ConsoleOne®, or NetWare® Administrator. Using a sample login script can help eliminate syntax errors, reducing the time it takes you to create your login scripts.

This chapter includes the following information:

- ◆ “Sample Container Login Script” on page 21
- ◆ “Sample Profile Login Script” on page 22
- ◆ “Sample User Login Scripts” on page 22
- ◆ “Sample Default Login Script” on page 23

For more information on how to create login scripts, see “Creating or Modifying Login Scripts” on page 13.

**IMPORTANT:** You must modify these sample login scripts to match the server names, directory paths, and specifications of your own network.

### Sample Container Login Script

The container login script should contain any information that applies to all users. By using some or all of the commands included in the following sample container login script, you might find it easier to plan and create your own container login scripts.

For more detailed information about each type of login script command, see “Login Script Commands and Variables” on page 25.

```
MAP DISPLAY OFF
MAP ERRORS OFF
MAP *1:=%HOME_DIRECTORY

IF "%1"="ADMIN" THEN MAP *1:SYS:SYSTEM

IF OS="WIN95" THEN
  MAP P:=SYS:PUBLIC
ELSE
  MAP INS S1:=SYS:PUBLIC
  MAP INS S2:=SYS:PUBLIC\%MACHINE%\OS\%OS_VERSION
END

IF OS="WINNT" THEN
  MAP P:=SYS:USERS\%LOGIN_NAME\WINNT
  MAP INS S16:=SYS:APPS\WINAPPS\WINNT
  SET TEMP = "P:\USERS\%LOGIN_NAME\WINNT\TEMP"
END

MAP INS S16:=VOL1:APPL\LOTUS
```

```

MAP INS S16:=SYS:EMAIL
MAP M:=VOL1:SHARED
MAP N:=VOL1:MODEMS

MAP O=SYS:DOC

IF MEMBER OF "MANAGERS" THEN
    MAP *3:=VOL1:PROJECTS\REPORTS
END

```

## Sample Profile Login Script

If you have groups of users with identical login script needs, you can create a Profile object and then create a login script for it. Then you can assign each user to be a member of that object.

The following login script shows an example of a profile login script you might create for users in the Profile object ACCOUNTING. This profile login script would execute after the container login script had executed.

For more detailed information about each type of login script command, see [“Login Script Commands and Variables” on page 25](#).

```

MAP DISPLAY OFF
MAP ERRORS OFF
MAP INS S16:=VOL1:APPL\DB
MAP *5:=VOL1:ACCOUNTS\NEW
MAP *6:=VOL1:ACCOUNTS\RECORDS

@GRPWISE5
MAP DISPLAY ON
MAP
WRITE
IF DAY_OF_WEEK="FRIDAY" THEN
    WRITE "Weekly progress report is due today."
    FIRE 2
END

```

## Sample User Login Scripts

The following login script is an example of a login script created for user MARY. The user login script executes after the container and profile login scripts have executed. In addition, a user login script prevents the default login script from executing for this specific user.

Therefore, you should consider whether any commands in the default login script are necessary for setting the user’s environment. If so, determine whether those commands should be placed into one of the other login scripts (container, profile, or user).

```

MAP DISPLAY OFF
MAP ERRORS OFF
MAP *7:=VOL1:MARY\PROJECTS\RESEARCH
MAP *8:=VOL1:FORMS
REM Mary needs access to FORMS while she’s on the
REM troubleshooting team.
SET WP="/u-mjr/b-5"
SET USER="mrichard"
#SEND /A=N

```

For more detailed information about each type of login script command, see [“Login Script Commands and Variables” on page 25](#).

## Sample User Login Script for Mapping Drives

In addition to creating User login scripts in ConsoleOne, users can also add modify their own user login scripts from the N menu in the system tray. This is very useful if users want to add commands to their login scripts to map drives that they access frequently. For more information on editing user login scripts from the N menu, see [“Modifying User Login Scripts from the N Menu” on page 17](#).

The following sample shows how Takashi included his drive mappings in his User login script.

```
MAP F:=\\MKTG.TOKYO.DIGITALAIRLINE.COM\DATA
MAP G:=\\MKTG.TOKYO.DIGITALAIRLINE.COM\HOME\KYAMATO
```

## Sample Default Login Script

The default login script executes the first time User object ADMIN logs in. It also executes for any users who do not have user login scripts.

You can't modify the default login script because it is coded into the Novell LOGIN utility. Instead, you can create container, profile, or user login scripts.

The following sample displays the contents of a default login script.

```
MAP DISPLAY OFF
MAP ERRORS OFF
MAP *1:=%FILE_SERVER\SYS:
MAP *1:=%FILE_SERVER\SYS:%LOGIN_NAME
IF "%LOGIN_NAME"="SUPERVIS" OR "%LOGIN_NAME"="ADMIN" THEN
MAP *1:=%FILE_SERVER\SYS:SYSTEM
MAP INS S1:=%FILE_SERVER\SYS:PUBLIC
MAP INS S2:=%FILE_SERVER\SYS:PUBLIC\%MACHINE%\OS\%OS_VERSION
MAP DISPLAY ON
MAP
```

For more detailed information about each type of login script command, see [“Login Script Commands and Variables” on page 25](#).





# 3

## Login Script Commands and Variables

This chapter lists login script commands alphabetically and by the type of tasks the commands perform. It also contains a list of identifier variables used in login scripts.

This chapter includes the following information:

- ◆ “[Login Script Conventions](#)” on page 25
- ◆ “[Using Identifier Variables](#)” on page 27
- ◆ “[Login Scripts Commands by Task](#)” on page 30
- ◆ Login Scripts Commands by Name (see specific commands)

Before implementing any commands or variables, make sure you understand basic login script conventions. See “[Login Script Conventions](#)” on page 25 and “[Using Identifier Variables](#)” on page 27.

**NOTE:** Commands that work only on DOS or Windows 3.x workstations have not been included. These commands include CLS, COMSPEC, DOS VERIFY, MACHINE, NOSWAP, and SWAP. For information on using these commands, see the help file that accompanies the Novell® Client™ for DOS and Windows\* 3.x software.

### Login Script Conventions

The following table explains the conventions you should follow when creating login scripts.

Convention	Explanation
Minimum login script	No minimum. All types of login scripts are optional. Login scripts can vary from one line to many. There are no required commands.
Case	Either uppercase or lowercase is accepted. Exception: identifier variables enclosed in quotation marks and preceded by a percent sign (%) must be uppercase. See “ <a href="#">Using Identifier Variables</a> ” on page 27.
Characters per line	Maximum 512 characters per line, including any variables after they are replaced by their values; 78 characters per line (common screen width) is recommended for readability.
Punctuation and symbols	Type all symbols (#, %, ", _) and punctuation exactly as shown in examples and syntax.
Commands per line	Use one command per line. Start each command on a new line; press Enter to end each command and start a new command.  Lines that wrap automatically are considered one command.

Convention	Explanation
Sequence of commands	<p>Generally, list commands in the order that you want them to execute, with the following restrictions:</p> <ul style="list-style-type: none"> <li>♦ ATTACH commands (NetWare® 3 only) must precede related MAP commands to avoid prompting the user for a username/password during login (see “ATTACH” on page 33).</li> <li>♦ If you use # (see “#” on page 32) or @ (see “@” on page 33) to execute an external program, the command must follow any necessary MAP commands. If sequence is not important, group similar commands, such as MAP and WRITE commands, together to make the login script easier to read.</li> </ul>
Blank lines	Blank lines don’t affect login script execution. Use them to visually separate groups of commands.
Remarks	<p>Lines beginning with REMARK, REM, an asterisk (*), or a semicolon (;) are comments, which don’t display when the login script executes (see “REMARK” on page 44).</p> <p>Use remarks to record the purpose of each command or group of commands or to temporarily keep certain lines from executing.</p>
Identifier variables	Type identifier variables exactly as shown. To display the value of an identifier variable as part of a WRITE command, you must enclose the identifier in quotation marks and precede it by a percent sign (%). See “Using Identifier Variables” on page 27.
eDirectory™ attribute values	<p>Any eDirectory attribute value, including extended names, can be read from a login script. The syntax for accessing eDirectory attributes is identical to common script variables with a few exceptions:</p> <ul style="list-style-type: none"> <li>♦ The eDirectory attribute must be at the end of the string.</li> <li>♦ If multiple variables are required, they must be in separate strings.</li> <li>♦ You must use the actual eDirectory attribute value names. You cannot use localized names or nicknames. Some common eDirectory attribute names are listed in “Using Identifier Variables” on page 27.</li> <li>♦ You must have the Read right to read the value of objects other than values associated with your own User object.</li> </ul> <p>If the name contains a space, you can replace it with an underscore (_) but it is not required. Both of the following examples are correct syntax:</p> <pre>map n %home directory map n %home_directory</pre>

Convention	Explanation
eDirectory object mappings	<p>eDirectory objects, such as cluster-enabled volumes, can be mapped in the login script using the object's fully distinguished name and context preceded by a leading dot (.).</p> <p>To map to a cluster-enabled volume, you would use the following syntax:</p> <pre>MAP N:=.eDirectory_object's_fully_distinguished_name_including_context</pre> <p>For example:</p> <pre>MAP N:=.cluster2_vol1.xyz.provo.novell</pre> <p>For more information on mapping, see <a href="#">"MAP" on page 41</a>.</p>

## Using Identifier Variables

Many login script commands allow you to take advantage of identifier variables to make login scripts more efficient and flexible.

Use identifier variables to enter a variable (such as `LAST_NAME`) rather than a specific name (such as `Smith`) in a login script command. When the login script executes, it substitutes real values for the identifier variables.

By using the variable, you can make the same login script command applicable to multiple users.

When using variables within login script commands, observe the following conventions:

- ◆ Type the variable exactly as shown.
- ◆ To use environment variables as identifiers, enclose them in angle brackets (`<>`).
- ◆ Identifier variables are used most often with commands such as `IF...THEN`, `MAP`, and `WRITE`. They can also be used with commands that you can specify a path for, such as `COMSPEC`.
- ◆ Identifier variables can be placed within literal text strings in a `WRITE` statement. However, the identifier variable must be in uppercase letters and preceded by a percent sign (%). (Literal text is the text that is displayed on the screen, such as "Sales report is due today." Literal text must be enclosed in quotation marks.)

For example, using the `%LAST_NAME` variable substitutes the user's actual last name for the `LAST_NAME` variable in the command when the user logs in. For example:

```
WRITE "HELLO, "%LAST_NAME
```

displays the following message on Bob Smith's workstation screen when he logs in:

```
Hello, SMITH
```

Using the `%GREETING_TIME` variable displays the time of day. If Bob Smith logs in during the morning, both of the following lines display the same message (Good morning, SMITH) on his screen:

```
WRITE "Good "; GREETING_TIME; ", "; LAST_NAME
```

```
WRITE "Good %GREETING_TIME,%LAST_NAME"
```

The following table lists the identifier variables.

Category	Identifier Variable	Function
Date	DAY	Day number (01 through 31)
	DAY_OF_WEEK	Day of week (Monday, Tuesday, etc.)
	MONTH	Month number (01 through 12)
	MONTH_NAME	Month name (January, February, etc.)
	NDAY_OF_WEEK	Weekday number (1 through 7; 1=Sunday)
	SHORT_YEAR	Last two digits of year (99, 00, 01)
	YEAR	All four digits of year (1999, 2000, 2001)
DOS Environment	<i>variable</i>	Any DOS environment variable can be used in angle brackets. To use a DOS environment variable in MAP, COMSPEC, and FIRE PHASERS commands, add a percent sign (%) in front of the variable.
Miscellaneous	ACCESS_SERVER	Shows whether the access server is functional (TRUE = functional; FALSE = nonfunctional)
	ERROR_LEVEL	An error number (0 = no errors)
	<i>%n</i>	Replaced by parameters the user enters at the command line with the login utility  See <a href="#">"Login Parameters with %n Variables" on page 30.</a>
NetWare Mobile™	DIALUP	0 = Not using dialup 1 = Using NetWare Mobile Client dialup profile
		OFFLINE
Network	FILE_SERVER	NetWare server name
	NETWORK_ADDRESS	Internal number assigned by the network specifying where a device can be located in the network cabling system
Object Properties	<i>property_name</i>	You can use property values of eDirectory objects as variables. Use the property values just as you do any other identifier variable. If the property value includes a space, enclose the name in quotation marks.  To use a property name with a space within a WRITE statement, you must place it at the end of the quoted string:  WRITE "Given name=%GIVEN_NAME" IF "MESSAGE SERVER"="MS1" THEN MAP INS S16:=MS1\SYS:EMAIL END

Category	Identifier Variable	Function
Time	AM_PM	a.m. or p.m.
	GREETING_TIME	Time of day (morning, afternoon, evening)
	HOUR	Hour (12-hour scale; 1 through 12)
	HOUR24	Hour (24-hour scale; 00 through 23)
	MINUTE	Minute (00 through 59)
	SECOND	Second (00 through 59)
User	CN	User's full login name as it exists in eDirectory
	LOGIN_ALIAS_CONTEXT	"Y" if REQUESTER_CONTEXT is an alias
	FULL_NAME	User's unique username. It is the value of the FULL_NAME property for both eDirectory and bindery-based NetWare. Spaces are replaced with underscores ( _ ).
	LAST_NAME	User's surname in eDirectory or full login name in bindery-based NetWare
	LOGIN_CONTEXT	User's context
	LOGIN_NAME	User's unique login name (long names are truncated to eight characters)
	MEMBER OF "group"	Group object that the user is assigned to
	NOT MEMBER OF "group"	Group object that the user is not assigned to
	PASSWORD_EXPIRES	Number of days before password expires
	REQUESTER_CONTEXT	Context when login started
USER_ID	Number assigned to each user	
Workstation	MACHINE	Type of computer (IBM_PC, etc.)
	NETWARE_REQUESTER	Version of the NetWare Requester™ for OS/2* or VLM™ users
	OS	Type of operating system on the workstation
	OS_VERSION	Operating system version on the workstation
	P_STATION	Workstation's node number (12-digit hexadecimal)
	PLATFORM	Workstation's operating system platform
	SHELL_TYPE	Version of the workstation's DOS shell; supports NetWare 2 and NetWare 3 shells and NetWare 4 Requester for DOS.

Category	Identifier Variable	Function
	SHELL_VERSION	The version of the Novell Client, as displayed in the property pages. Provides a more effective testing of the Novell Client version from the login script.
	SMACHINE	Short machine name (IBM*, etc.)
Workstation (cont'd)	STATION	Workstation's connection number
	WINVER	Version of the workstation's Windows operating system

## Login Parameters with %n Variables

Some variables in a login script can be indicated by a percent sign (%) followed by a number from 0 through 9.

**NOTE:** Only four variables (%2 through %5) can be changed in the login screen.

When a user logs in, additional parameters can be entered that the LOGIN utility passes to the login script. The utility then substitutes these parameters for any %n variables in the login script. These variables are replaced in order by the parameters the user entered when logging in.

The %0 variable is replaced by the name of the NetWare server entered in the Login dialog box, and %1 is replaced by the user's fully distinguished login name. The remaining variables change, depending on what the user types when executing the LOGIN utility. The %n variables must precede all command line options.

The **SHIFT** (page 46) command allows you to change the order in which these %n variables are substituted.

## Login Scripts Commands by Task

This list identifies which specific login script commands to use, depending on what you need your login scripts to accomplish.

### Network Connection and Resource Access

Action	Command
Establish a connection between a workstation and a NetWare server	"ATTACH" on page 33
Set a user's current context in the eDirectory tree	"CONTEXT" on page 34
Change the default drive while the login script is executing	"DRIVE" on page 35
Map drives and search drives to network directories or to map to eDirectory objects such as cluster-enabled volumes	"MAP" on page 41
Attach to another eDirectory tree within the network and access its resources	"TREE" on page 48

## Login Script Execution

Action	Command
Allow the user to terminate execution of the login script	<a href="#">“BREAK” on page 34</a>
Emit a phaser sound or beep	<a href="#">“FIRE or FIRE PHASERS” on page 36</a>
Execute a portion of the login script out of the regular sequence	<a href="#">“GOTO” on page 37</a>
Perform an action only under certain conditions	<a href="#">“IF...THEN” on page 37</a>
Execute independent files or another object’s login script as a part of the login script currently being processed	<a href="#">“INCLUDE” on page 40</a>
Prevent the default user login script from running and do not create any user login scripts in a container or profile login script	<a href="#">“NO_DEFAULT” on page 43</a>
Create a pause in the execution of the login script	<a href="#">“PAUSE” on page 43</a>
Set or override a user’s assigned or command line-specified profile script in a container script	<a href="#">“PROFILE” on page 43</a>
Set a home server where the bindery login script is read from	<a href="#">“SCRIPT_SERVER” on page 44</a>
Change the order in which %n variables are interpreted in the login script	<a href="#">“SHIFT” on page 46</a>

## Workstation Environment

Action	Command
Set an environment variable to a specified value	<a href="#">“SET” on page 45</a>
Set the workstation time equal to the time on the NetWare server that the workstation first connects to	<a href="#">“SET_TIME” on page 46</a>

## Text File Usage

Action	Command
Execute a program that is external to the login script	<a href="#">“#” on page 32</a>
Execute a program that is external to the login script and then continue with the script	<a href="#">“@” on page 33</a>
Show the contents of a text file when the user logs in	<a href="#">“DISPLAY” on page 34</a>
Terminate execution of the login script	<a href="#">“EXIT” on page 35</a>
Show the text of a word processing file when the user logs in	<a href="#">“FDISPLAY” on page 36</a>
Stop the login script and return an error code	<a href="#">“TERM” on page 47</a>
Display messages on the workstation screen when a user logs in to the network	<a href="#">“WRITE” on page 49</a>

## Other

Action	Command
Display the last time the user logged in	<a href="#">LASTLOGINTIME (page 41)</a>
Include explanatory text in the login script or to keep a line from being executed during testing	<a href="#">REMARK (page 44)</a>

## #

Use the # command to execute a program that is external to the login script. The # command executes an external program and waits until it is finished running before continuing with other login script commands.

This command fails when

- ◆ The given directory is invalid
- ◆ Proper security rights are lacking
- ◆ The executable file cannot be found
- ◆ Insufficient workstation memory is available to load the file

**IMPORTANT:** Use the @ command instead of the # command to run an external program from a login script if that external program will remain open for any length of time. Otherwise, the login script will remain open until that external program is closed. (Login scripts cannot be edited while they are open.)

## Command Format

```
# [path] filename [parameter]
```

Replace *path* with a drive letter or, if you have specified NOSWAP on the command line or in the login script, you can replace variable with a full directory path beginning with the NetWare volume name.

Replace *filename* with an executable file (files that end in .EXE, .COM, or .BAT, for example). It isn't necessary to include the extension, but doing so can speed up the execution of the external program.

Replace *parameter* with any parameters that must accompany the executable file.

## Example

You can use Automatic Client Update (ACU) to update Novell Client software without having to physically visit each workstation and manually update the software.

ACU uses a text file that can be called from the login script. To call this file from the login script for a workstation running Novell Client for Windows NT/2000, place one of the following commands in the login script:

```
#Z:\PUBLIC\INSTALL\NTCLIENT\SETUPNW.EXE /ACU
```

or

```
#\[SERVER]\[VOLUME]\PUBLIC\INSTALL\NTCLIENT\SETUPNW.EXE /ACU
```



# @

Use the @ command to execute a program that is external to the login script and then continue with the script (similar to the Startup group). Enter the @ command followed by the name of the file that you want to execute.

**IMPORTANT:** Use the @ command instead of the # command to run an external program from a login script if that external program will remain open for any length of time. Otherwise, the login script will remain open until that external program is closed. (Login scripts cannot be edited while they are open.)

This command fails when

- ◆ The given directory is invalid
- ◆ Proper security rights are lacking
- ◆ The executable file cannot be found
- ◆ Insufficient workstation memory is available to load the file

## Command Format

```
@ [path] filename [parameter]
```

Replace *path* with a drive letter.

Replace *filename* with an executable file (files that end in .exe, .com, or .bat, for example). Do not include the extension.

Replace *parameter* with any parameters that must accompany the executable file.

## Examples

If you want to start the GroupWise® program from within the login script, you must have a search drive mapped to where the GroupWise application is stored. Enter the following command in the login script:

```
@GRPWISE5
```

You do not need to enter a path in this case, because the executable program file for GroupWise 5 (grpwise5.exe) is located in a search drive.

If you do not have a search drive mapped to the directory where the program is located, include the path to that directory in the command. For example, if the program to run GroupWise 5 is located in the APPS directory on drive G:, use the following command:

```
@SYS:\APPS\GRPWISE5
```

## ATTACH

Establishes a connection between a workstation and a NetWare server. If the server is not in the current tree, a bindery connection is made.

In networks running NetWare 3 or earlier versions, use the ATTACH command to connect to multiple file servers. In NetWare 4 and later, users no longer need to attach separately to multiple servers and do not need to use the ATTACH command. If users want to connect to multiple trees, use the TREE command (see “TREE” on page 48).

**TIP:** To create persistent drive mappings, use the MAP command (see “MAP” on page 41).

# BREAK

Use `BREAK ON` to allow the user to terminate execution of the login script. The default is `BREAK OFF`. If `BREAK ON` is included in a login script, the user can press `Ctrl+C` or `Ctrl+Break` to abort the normal execution of the login script.

**NOTE:** When the `BREAK` option is `ON`, type-ahead keyboard input is not saved to the buffer.

## Command Format

```
BREAK ON|OFF
```

# CONTEXT

Use `CONTEXT` to set a user's current context in the eDirectory tree. Similar to the `CX` utility in DOS, you can use a complete name to move down through the context, or you can use periods to move up toward the root of the tree. However, `CONTEXT` does not support all options that the `CX` workstation utility does; `CONTEXT` only sets the context.

## Command Format

```
CONTEXT context
```

To change the current eDirectory context, replace *context* with the context that you want the user to see after login.

## Example

To change the context to the Organizational Unit `SALES`, under the Organization `NOVELL_US` add the following line to the login script:

```
CONTEXT .SALES.NOVELL_US
```

You can type a single period instead of a container name to indicate that you want to move up one level.

For example, if you are in the context `SALES.NOVELL_US` and you want to move up one level to the context `NOVELL_US`, add the following line to the login script:

```
CONTEXT .
```

To move up two levels, enter two periods, etc.

# DISPLAY

Use `DISPLAY` to show the contents of a text file when the user logs in.

When you use `DISPLAY`, all characters in the file, including any printer and word processing codes, appear. This command works best with an ASCII file.

To display only the text and suppress codes, use [“FDISPLAY” on page 36](#).

**NOTE:** If the given path does not exist or if the file is not found, no error message appears on the screen when the user logs in.

## Command Format

```
DISPLAY [path] filename
```

Replace *path* with either a drive letter or a full directory path beginning with the NetWare volume name.

Replace *filename* with the complete name (including the extension) of the file that you want to display.

## Example

Suppose you put messages in a file called `sysnew.txt` in the `sys:public\messages` directory, and you want your users to see the messages when they log in on Monday. Add the following lines to the container login script:

```
IF DAY_OF_WEEK="Monday" THEN
DISPLAY SYS:PUBLIC\MESSAGES\SYSNEWS.TXT
END
```

## DRIVE

Use `DRIVE` to change the default drive while the login script is executing.

If this command is not included in the login script, the default drive will be set to the first network drive, which is often assigned to the user's home directory upon login.

If you don't want the default drive to be the first network drive, map a drive in the login script to the directory you want to be the default; then use the `DRIVE` command to change the default drive.

Instead of specifying a drive letter such as `F:` or `G:`, you can use an asterisk followed by a number *n* to represent the *n*th network drive (for example, `*3`). This allows drive letters to reorder themselves automatically if previous drive mappings are deleted or added.

## Command Format

```
DRIVE [drive | *n]
```

Replace *drive* with a local or network drive letter, or replace *n* with a drive number. The use of either is dependent on their already being assigned within the login script.

## Example

Suppose a user will be working on only one project for several days and the files for that project are located on drive `S:`. Use the `DRIVE` command to set the default drive to `S:` so that the user doesn't have to change the default drive manually after each login.

First, make sure drive `S:` is mapped to the correct directory in the user's login script. Then add the following command to the login script:

```
DRIVE S:
```

## EXIT

Use `EXIT` to terminate execution of the login script.

**IMPORTANT:** You cannot use EXIT in a login script to stop the login script and execute a program. EXIT only terminates the execution of the login script. If you want to execute a program after exiting the login script, you must use one of two commands: “#” on page 32 or “@” on page 33 followed on the next line by EXIT.

## FDISPLAY

Use FDISPLAY to show the text of a word processing file when the user logs in.

To display both the text and the printer and word processing codes of a file, or to display an ASCII file, see “DISPLAY” on page 34.

When you use FDISPLAY to display a word processing file, the text is filtered and formatted so that only the text is displayed. FDISPLAY will not display tabs.

**NOTE:** If the given path does not exist or if the file is not found, no error message appears on the screen when the user logs in.

### Command Format

```
FDISPLAY [path] filename
```

Replace *path* with either a drive letter or a full directory path beginning with the NetWare volume name.

Replace *filename* with the complete name (including the extension) of the file that you want to display.

### Examples

Suppose you put messages in a file called sysnews.txt in the sys:public\messages directory, and you want your users to see this file on their screens when they log in on Monday.

Add the following lines to the container login script:

```
IF DAY_OF_WEEK="Monday" THEN  
FDISPLAY SYS:PUBLIC\MESSAGES\SYSNEWS.TXT  
END
```

## FIRE or FIRE PHASERS

FIRE or FIRE PHASERS emits a long beep or a phaser sound.

Use this command to generate the phaser sound whenever a user logs in. Use FIRE or FIRE PHASERS with the IF...THEN command to make the sound execute a different number of times depending on the circumstances of the login.

### Command Format

```
FIRE n soundfile
```

Replace *n* with the number of times that you want this sound to occur.

Replace *soundfile* with the name of the sound file that you want to play when this command is executed. (The sound file variable cannot be used on DOS workstations.) You can use any .wav or platform-compatible sound file.

## Examples

The following line executes the phaser sound four times upon login:

```
FIRE 4
```

The following line executes the rifle sound three times upon login:

```
FIRE 3 RIFLE.WAV
```

To use a variable as the number of times to fire, use % before the variable, as follows:

```
FIRE %variable
```

For more information about using variables, see [“Using Identifier Variables” on page 27](#).

## GOTO

Use GOTO to execute a portion of the login script out of the regular sequence.

Set BREAK ON in your login script before experimenting with GOTO loops so that you can break out of a login script if necessary. See [“BREAK” on page 34](#).

**IMPORTANT:** Do not use GOTO to enter or exit a nested [IF...THEN \(page 37\)](#) statement. This usage causes problems for the program.

## Command Format

```
GOTO label
```

Use *label* to indicate where the login script should continue executing.

## Example

To execute a loop of commands, include the following lines in the login script. In this case, the commands to be executed are labeled AGAIN (as indicated in the second line).

```
SET X="1"  
AGAIN:  
SET X=<X> + "1"  
;see compound strings for this  
WRITE <X>  
IF <X> < "9" THEN GOTO AGAIN
```

The GOTO command looks at the value of <X> (a DOS environment variable). If the value of <X> is less than 9, then <X> increments by 1 and GOTO loops back to the AGAIN label. When <X> gains the value of 9, the IF...THEN test becomes false, the GOTO is ignored, and the script continues normally.

## IF...THEN

Use IF...THEN to perform an action only under certain conditions.

An example of a conditional statement is

```
IF MEMBER OF "CLERKS"
```

In this statement, some action is performed if the user who logged in belongs to the Group object named CLERKS.

The following is a different type of conditional statement:

```
IF DAY_OF_WEEK="MONDAY"
```

In this statement, the equal sign (=) indicates the relationship between the variable (DAY\_OF\_WEEK) and its value (Monday). Note that the value (Monday) is inside quotation marks.

When using IF...THEN statements, be aware of the following syntax rules:

- ◆ Use AND or OR to include two or more conditionals in an IF...THEN statement.
- ◆ Values of conditional statements must be enclosed in quotation marks.
- ◆ Values of conditional statements are compared with the assumption that the values are characters, not numeric values. The value of 21, therefore, would be considered greater than the value of 100 when comparing these two characters. To ensure the system properly calculates numeric values instead of character values, use the VALUE modifier in the IF...THEN statement.
- ◆ The ELSE statement is optional.
- ◆ IF, ELSE, and END must be on separate lines. THEN does not need to be on a separate line.
- ◆ If you include a WRITE command as part of the IF...THEN command, the WRITE command must be on a separate line.
- ◆ IF...THEN statements can be nested (up to 10 levels). However, GOTO should not be used in a nested IF...THEN statement to enter or exit from the body of an IF...THEN statement.
- ◆ If your IF...THEN statement consists of only one line, you do not need to include END even if that line wraps. If your IF...THEN statement must be on more than one line (for example, if you used ELSE or WRITE, which must be on separate lines), you must include END.
- ◆ Six relationships are possible between the elements of an IF...THEN statement. Represent these relationships with the following symbols:
  - = Equals
  - <> Does not equal
  - > Is greater than
  - >= Is greater than or equal to
  - < Is less than
  - <= Is less than or equal to

## Command Format

```
IF conditional [AND|OR [conditional]] THEN  
commands  
[ ELSE  
command]  
[ END ]
```

Replace *conditional* with identifier variables. For information about identifier variables, see [“Using Identifier Variables” on page 27](#).

Replace *commands* with any login script commands that you want to be executed if the specified condition is true.

## Examples

If you place the following command in a login script, the message `Status report is due today` appears when the user logs in on Monday and `Have a nice day!` on other days:

```
IF DAY_OF_WEEK="MONDAY" THEN
    WRITE "Status report is due today."
ELSE
    WRITE "Have a nice day!"
END
```

The following lines mean “If the hour (on a 24-hour scale) is greater than or equal to 12, then write ‘afternoon’”:

```
IF VALUE HOUR24>="12" THEN
    WRITE "afternoon"
END
```

The following command executes the `CAPTURE` utility on the fourth day of the week (Wednesday):

```
IF NDAY_OF_WEEK="4" THEN
    #CAPTURE Q=FAST_Q NB TI=10 NFF
END
```

The following example shows nested `IF...THEN` statements. Notice that there are two `IF` statements, so each one must have its own `END` statement:

```
IF DAY_OF_WEEK="MONDAY" THEN
    MAP *6:=VOL1:APPL\WP
    IF MEMBER OF CLERKS THEN
        WRITE "Your report is due immediately!"
    END
END
```

Conditionals can be joined with commas, the word `AND`, or the word `OR` to form compound conditionals.

The first line of the following `IF...THEN` statement is a compound conditional that means “If it is the evening of the first day of the month”:

```
IF GREETING_TIME="EVENING" AND DAY="01" THEN
    WRITE "The system will be backed up tonight."
END
```

An `IF...THEN` statement can include several commands that must be executed if the conditional is true.

The following example shows two commands that are executed on Tuesdays: a `WRITE` command that displays a message about a staff meeting, and an `INCLUDE` command that tells the login script to process any commands or messages contained in the file `sys:public\update.txt`.

```
IF DAY_OF_WEEK="TUESDAY" THEN
    WRITE "Staff meeting today at 10 a.m."
    INCLUDE SYS:PUBLIC\UPDATE.TXT
END
```

# INCLUDE

Use INCLUDE to execute independent files or another object's login script as a part of the login script currently being processed.

These subscripents can be text files that contain valid login script commands (any of the commands explained here) or login scripts that belong to a different object you have rights to.

Text files that contain login script commands, as well as other objects' login scripts, can be used as subscripents. Use these subscripents to supplement the main login script.

You can create and edit text file subscripents using any text editor. Subscripents do not need to have any particular filenames or extensions.

The INCLUDE command executes the login script commands contained in the subscript. It does not display the text of the subscripents.

INCLUDE nesting is limited only by available memory. This means that one subscript file can include another subscript file, which can include yet another subscript file, etc. In DOS, however, the maximum number of subscript files that you can nest is 10.

If the subscript is a text file, users must have at least File Scan and Read rights to the directory containing the subscript.

If you are using another object's login script as a subscript, users must have the Browse right to the object whose script you are including and the Read right to the object's Login Script property.

## Command Format

**INCLUDE** [*path*]*filename*

or

**INCLUDE** *object\_name*

To use a text file as a subscript, replace *path* with either a drive letter or a full directory path beginning with the NetWare volume name.

Replace *filename* with the complete name (including the extension) of the text file.

To execute another object's login script as part of a login script, replace *object\_name* with the name of the object whose login script you want to use.

## Examples

To execute a text file called script.new (located in volume vol1) as a subscript, add the following line to your main login script:

```
INCLUDE VOL1:ADMIN\USERS\SCRIPT.NEW
```

Suppose you are creating a container login script for all users under the Organizational Unit object SALES\_LA. You recently created a container login script for users under the Organizational Unit object SALES\_PV.

You've decided that the SALES\_LA users could use the same login script as the SALES\_PV users, but with a few more drive mappings.

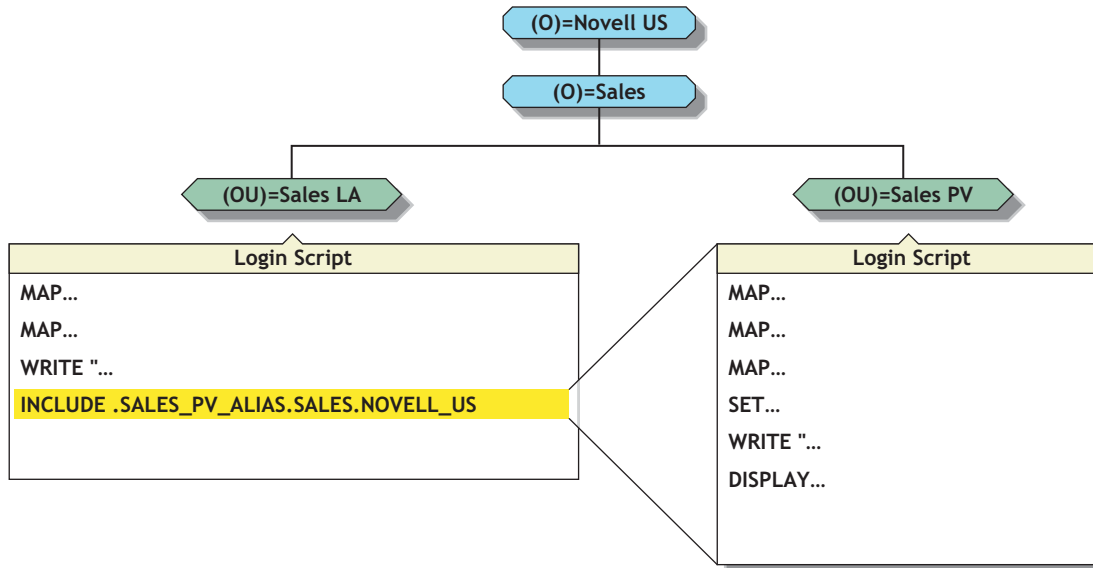


In the SALES\_LA login script, you could add the additional drive mappings and then use the INCLUDE command to execute the entire SALES\_PV login script as a part of the SALES\_LA login script, as follows:

1. Create an alias for the SALES\_PV Organizational Unit in the SALES\_LA Organizational Unit.
2. Add this line to the SALES\_LA Organizational Unit's login script:

```
INCLUDE .SALES_PV_ALIAS.SALES.NOVELL_US
```

The following figure illustrates how the INCLUDE command executes the SALES\_PV login script as part of the SALES\_LA login script.



## LASTLOGINTIME

Use LASTLOGINTIME to display the last time the user logged in.

If you include this command in the login script, the time of the last login is displayed on the workstation screen.

## Command Format

```
LASTLOGINTIME
```

## MAP

Use MAP to map drives and search drives to network directories or to map to eDirectory objects such as cluster-enabled volumes.

If you use MAP to automate drive map assignments during execution of the login script, users don't need to map drives every time they log in.

If you do not want the result of each mapping to be displayed as it is executed, add the MAP DISPLAY OFF command at the beginning of the login script. When all drive map assignments

have been completed, add the line `MAP DISPLAY ON` and `MAP` to your login script. This sequence provides a cleaner display for users as they log in.

Instead of specifying drive letters such as F: or G:, you could use an asterisk followed by a number *n* to represent the *n*th network drive. For example, if the first network drive is F:, then using `MAP *3:=` would assign H: {1 2 3 = F G H}. Or, if the first network drive is D:, then using `MAP *4:=` would assign G: {1 2 3 4 = D E F G}.

This allows drive letters to reorder themselves automatically when local drives are removed or added or when the first network drive is changed. This also allows users to log in from workstations with a different number of local drives than their regular workstation.

You can map a local drive (usually A: through C:) to a network directory, but you cannot access the local drive until you remove the network drive mapping. Do not map over the local drive containing the Windows directory.

Do not map a drive to a network drive such as a CD-ROM drive.

## Command Format

**MAP** [*options*] | [*parameters*][*drive:=path*]

Replace *drive* with any valid network drive letter, local drive letter, or search drive number.

Replace *path* with either a drive letter, a full directory path, a Directory Map object, or an eDirectory object such as a cluster-enabled volume.

More than one command can be on the map line if the commands are separated by a semicolon (;), as shown in the following example:

```
MAP *1:=SYS:PUBLIC;*2:=SYS:PUBLIC\DOS
```

When mapping a drive to a directory on an eDirectory server, begin the path with either the Volume object name or *server\volume*.

When mapping to a directory on a bindery-based server or to an eDirectory server that isn't the current server, begin the path with the server's name.

When mapping to an eDirectory object, use the object's fully distinguished name preceded by a leading period (.).

Replace *option* with one of the following:

- ◆ `DISPLAY ON|OFF` determines whether drive mappings are displayed on the screen when the user logs in. The default setting is ON. This option is valid only in login scripts.
- ◆ `ERRORS ON|OFF` determines whether MAP error messages are displayed when the user logs in. `MAP ERROR OFF` must be placed before MAP commands in the login script. The default setting is ON. This option is valid only in login scripts.

Replace *parameter* with one of the following:

- ◆ `INS` inserts a drive mapping between existing search mappings.
- ◆ `DEL` deletes a drive mapping, making that drive letter available for other mapping assignments.
- ◆ `ROOT` or `R` maps a fake root. Windows NT\* and Windows 2000 are always mapped to the root. Some applications require their executable files to be located in a root directory.

Because you might not want users to have rights at the root directory, you can map a fake root to a subdirectory instead.

The Windows NT/2000 native environment forces a map root on all drives. To prevent a forced map root in a Windows NT/2000 environment, set the `MAP ROOT OFF = 1` environment variable. All drives are then mapped as specified, and only explicit map root drives are rooted.

- ◆ `C (CHANGE)` changes a search drive mapping to a regular mapping or a regular mapping to a search drive mapping.
- ◆ `P (PHYSICAL)` maps a drive to the physical volume of a server rather than to the Volume object's name.

A Volume object name might conflict with a physical volume name. (For example, object `ACCT` is an Accounting volume, but there is also a physical volume named `ACCT`.) Therefore, if you prefer to map a drive to the physical volume name, use `MAP P`.

- ◆ `N (NEXT)` maps the next available drive when used without specifying a drive number or letter.

## NO\_DEFAULT

Use `NO_DEFAULT` in a container or profile login script if you do not want the default user login script to run and you do not want to create any user login scripts.

**NOTE:** If you have created a user login script for a user, that login script executes whether or not the `NO_DEFAULT` command is in the container or profile login script.

## Command Format

`NO_DEFAULT`

## PAUSE

Use `PAUSE` to create a pause in the execution of the login script.

You can add `PAUSE` to the login script following a message so that the user has time to read the message before it scrolls off the screen.

If you include `PAUSE`, the message `Strike any key when ready...` appears on the workstation screen. NetWare Login then waits for a key to be pressed before it executes the rest of the login script.

## Command Format

`PAUSE`

## PROFILE

Use `PROFILE` in a container script to set or override a user's assigned or command line-specified profile script. This is useful when defining a group profile.

## Command Format

```
PROFILE profile_object_name
```

## Example

To override the profile script assigned to a user, and to cause the user to execute a PROFILE script called `team_profile`, use the following command:

```
PROFILE team_profile
```

## REMARK

To include explanatory text in the login script or to keep a line from being executed during testing, begin a line with REMARK, REM, an asterisk (\*), or a semicolon (;). Any text that follows these symbols is ignored when NetWare Login executes the login script. Remarks do not appear on the screen.

**IMPORTANT:** This command and its associated text must be the only entry on a line. Placing remarks on the same line as other login script commands can cause errors.

If a remark is several lines long, begin each line with the remark keyword.

## Command Format

```
REM[ARK] [text]
```

or

```
* [text]
```

or

```
;  
[text]
```

Replace text with the comment you want to include in the login script.

## Example

The following are examples of explanatory text that you might use with the REMARK command and its variants:

```
* This is Richard's login script  
; Mapped network drives follow:  
REM The next mapping is a fake root.  
REMARK This login script is for new users.
```

## SCRIPT\_SERVER

NetWare 2 and NetWare 3 users can use SCRIPT\_SERVER to set a home server where the bindery login script is read from.

SCRIPT\_SERVER has no effect on NetWare 4 and later users.

## Command Format

```
SCRIPT_SERVER server_name
```

## SET

Use SET to set an environment variable to a specified value.

When you use SET in a login script, you must include quotation marks (" ") around values.

If a variable is set to a path that ends in a backslash and quote ("\), these two characters are interpreted as an embedded quote preceded by an escape character. To avoid this problem, use two backslashes before the ending quotes (\\).

You do not need to include SET commands in login scripts. For example, you might decide to place some SET commands in the workstation's autoexec.bat file. Where you use SET commands depends upon your individual needs.

This command does not work in a login script if the DOS workstation's environment is too small. In this case, you should set the environment size in the CONFIG.SYS file.

After you use the SET command to set a value for an environment variable, you can use that variable in other login script commands.

To include an environment variable as an identifier variable in a command, enclose the name of the variable in angle brackets (for example, <emailuser>).

## Command Format

```
[TEMP] SET name="value"
```

Replace *name* with an environment parameter that identifies the environment you want to change.

Replace *value* with identifier variable substitutions. Values must be enclosed in quotation marks.

To change the environment for the login script, but not for the workstation itself after the login script has finished executing, use the optional keyword TEMP.

## Examples

You can use SET to make a prompt display the current directory path (such as F:\HOME\MARY>) rather than just the drive letter. To do this, add the following line to the login script:

```
SET PROMPT="$P$G"
```

"\$P" lists the current directory path and "\$G" displays a greater-than sign (>).

To set a path for a program called DAILY, which is in the REPORTS subdirectory under drive G:, you would add the following line:

```
SET PATH="G:\REPORTS\DAILY"
```

This sets the variable PATH to g:reports\daily.

Setting the variable PATH in the login script removes any search drives previously assigned. Use SET PATH only before you map search drives. SET PATH also overwrites any paths set in the user's autoexec.bat file.

To display this path, you can include PATH as an identifier variable in a WRITE command by enclosing the variable (not the value) in angle brackets. For example, the following line displays My path is G:\REPORTS\DAILY:

```
WRITE "My path is ";path
```

To include an environment variable in a MAP command, precede the variable with a percent sign (%). For example, you could include the following lines in a login script to set and map a drive to the variable NWS:

```
SET NWS="C:\XYZ"  
MAP S16:=%<NWS>
```

## SET\_TIME

Use SET\_TIME to set the workstation time equal to the time on the NetWare server that the workstation first connects to.

If you set SET\_TIME to OFF in the login script, the workstation time does not update to the server's time.

### Command Format

```
SET_TIME ON|OFF
```

## SHIFT

Use SHIFT to change the order in which %n variables are interpreted in the login script. SHIFT allows users to enter login parameters in any order. For more information on %n variables, see [“Login Parameters with %n Variables” on page 30](#).

You can shift up to 10 login script variables from %0 to %9.

When users execute NetWare Login, they can include additional parameters. Each of these parameters is assigned a %n variable; in this way, the parameter's real value can be substituted for the %n variable that appears in the login script.

In the login script, you can add SHIFT with a positive or negative number to move the variables in either direction. For example, SHIFT -3 moves each %n variable three positions to the right.

### Command Format

```
SHIFT [n]
```

Replace n with the number of places that you want the variable to shift. The default is SHIFT 1.

## Examples

When Mary logs in, she wants to access her word processing program, change the way it is set up, and map a drive to her work directory called ACCNTS.

Mary also has a command in her login script to map a drive to her DATABASE directory, but she does not need it today. The commands in Mary's login script are shown here:

```

:LOOP
IF "%2"="WP" THEN
    SET WP="\U-CML\B-10\D"
    MAP S16:=SYS:APPL\WP\SETUP
END
IF "%2"="ACCNTS" THEN MAP G:=SYS:ACCNTS
IF "%2"="DATABASE" THEN MAP S16:=SYS:APPL\DATABASE
SHIFT
IF "%2"<>" " THEN GOTO LOOP

```

(In the last line,

```
IF "%2" < >
```

is followed by closed quotation marks, which means “If %2 isn’t blank”).

With these commands in her login script, and assuming a command line login, Mary can log in to the primary file server (named FS1) using her username, MARY, as follows:

```
LOGIN FS1\MARY WP ACCNTS
```

The parameters in Mary’s login command are given the following values:

```

%0=FS1
%1=MARY
%2=WP
%3=ACCNTS

```

Mary’s login script looks for %2, which is WP, and sets the word processing environment. Then the login script shifts the variables one to the left so that %2 now becomes ACCNTS. Upon executing the loop, the login script maps a drive to the ACCNTS directory.

Mary could also change the order of her login command without affecting the way her work environment is set up, as follows:

```
LOGIN MARY ACCNTS WP
```

The parameters in this login command are given the following values:

```

%0=FS1
%1=MARY
%2=ACCNTS
%3=WP

```

In this case, Mary’s login script looks for %2, which is now ACCNTS. The login script maps a drive to the ACCNTS directory. Then the login script shifts the variables to the left so that %2 now becomes WP.

Upon executing the loop, the login script sets the word processing environment.

## TERM

This command is normally used only for Application Launcher scripts, a component of ZENworks® for Desktops.

You can use the TERM command in a login script to stop the login script and return an error code.

You can also use TERM in an IF...THEN statement, so that the login script stops and an error code is returned only if an IF statement is true (that is, a certain condition exists). If the IF statement is

false (that is, a condition doesn't exist), the login script skips the TERM command and continues executing.

Because TERM stops the login script, be sure to put this command either at the end of the login script or at a point within the script where you intend execution to stop. Do not nest the TERM command in the login script.

If you add TERM to a container login script, it prevents other profile or user login scripts from running. If you put TERM in a profile login script, it prevents the user login script from running.

## Command Format

```
TERM nnn
```

Replace *nnn* with an error level. Any error level from 000 through 999 is valid.

## Example

If you want the login script to exit and return an error code of 718, you can add the following statement to the login script:

```
TERM 718
```

## TREE

The TREE command can be used only with clients that support multiple eDirectory tree attachments.

Use TREE to attach to another eDirectory tree within the network and to access its resources.

The TREE command changes the “focus” of the login script so that all eDirectory object references in subsequent script commands (for drive mappings, print captures, etc.) apply to the eDirectory tree specified in the TREE command.

You can include multiple TREE commands within a login script, either to attach to additional trees or to switch the login script's “focus” back to a tree that the user is already attached to.

## Command Format

```
TREE tree_name[/complete_name;password]
```

Replace *tree\_name* with the name of the eDirectory tree that you want the user to attach to.

Replace *complete\_name* with the user's complete name (Distinguished Name) for the eDirectory tree that the user is attaching to. The complete name establishes the user's context in the tree. If you do not include the complete name, the user is prompted for a complete name when the TREE command is executed from the login script.

Replace *password* with the correct password for that user and tree. If the username and password are the same as the primary login username and password, you can omit the password and not be prompted for it.

**IMPORTANT:** Use caution when including passwords in a login script. It is more secure to eliminate the password. Then, at the point in the login script where the TREE command is executed, the user is prompted for the password.



## Example

To attach the user with the complete name MRICHARD.ACME (whose password is BUTTERFLY) to an eDirectory tree named CORP, add the following line to the login script:

```
tree corp/.mrichard.acme;butterfly
```

## WRITE

Use WRITE to display messages on the workstation screen when a user logs in to the network. Text that you want to display must be enclosed in quotation marks (" ").

There are several ways to display variables in the text message. The way you enter the variable in the WRITE command determines the display format, as follows:

- ◆ If you type the identifier variable as shown, with no special punctuation, only the variable is displayed on the screen.
- ◆ If you enclose the identifier variable inside quotation marks, precede the variable with a percent sign (%) and type it in uppercase letters. This method is often used to combine regular text with an identifier variable, because both the text and the variable can be enclosed in the same quotation marks.
- ◆ To join several text strings and identifier variables into a single display without enclosing the variables in quotation marks, use a semicolon (;) between the text and the variables.
- ◆ If you have several WRITE commands, each one appears on a separate line on the user's workstation. However, if you put a semicolon at the end of all but the last WRITE commands, the commands all appear as one continuous sentence or paragraph (although they might wrap onto additional lines on the workstation's screen).

Text strings can include the following special characters:

Character	Meaning
\r	Causes a carriage return
\n	Starts a new line of text
\"	Displays a quotation mark on the screen
\7	Makes a beep sound

In addition to the semicolon, you can use other operators to form compound strings (in other words, to join text and identifier variables into one command). These operators are listed in the following table, in order of precedence:

Operator	Meaning
* / %	Multiply, divide
+ -	Add, subtract
>> <<	Shift left or right (1000 >> 3 becomes 1)

## Command Format

```
WRITE "[text][%identifier]" [;][identifier]
```

Replace *text* with the words you want to display on the screen.

Replace *identifier* with a variable you want to display, such as a user's login name. See [“Using Identifier Variables” on page 27](#) for a complete list of variables.

## Examples

To display the message `Hello`, add the following line to the login script:

```
WRITE "Hello"
```

To display the user's surname along with a greeting, add the identifier `LAST_NAME` to the command. To do this, either join the text and the identifier with a semicolon (;) or include the variable in the quotation marks with the text.

For example, either of the following lines displays `Hello, Smith` when user Bob Smith logs in:

```
WRITE "Hello, ";%LAST_NAME
```

```
WRITE "Hello, %LAST_NAME"
```

To make a beep sound occur while the phrase `Good morning` appears on the screen, add the following line to the login script:

```
WRITE "Good %GREETING_TIME \7"
```

# 4

## Troubleshooting Login Scripts

Though login scripts are useful in setting up drive mappings, search mappings, the user's printing environment, and displaying important messages and other commands, they can sometimes be a source of frustration when they don't work as expected.

A majority of the problems that occur with login scripts are associated with mapping drives and capturing printers. Usually, these problems can be attributed to syntax and spelling errors. Correcting spelling errors and using variables in the right way resolve many problems. For example, you must put quotes around a login name in an IF..THEN state met.

### Locating the Problem Statement in the Login Script

Try the following to locate the problem statement in the login script:

- ◆ Set MAP DISPLAY and MAP ERRORS to ON in the login script.

By setting MAP DISPLAY and MAP ERRORS to ON, the errors will display, often providing clues as to why a certain command did not execute as expected. After seeing that the login script performs all map commands without error, the MAP DISPLAY and MAP ERRORS parameters can be turned OFF.

- ◆ Use the WRITE or MAP commands together with the PAUSE command prior to the script command which doesn't work as expected.

By adding the PAUSE command after the WRITE or MAP command, you have an opportunity to see what is happening when the script is executed.

For example, if the login script didn't seem to be completing an IF..THEN statement, put both a WRITE and a PAUSE statement immediately before the IF..THEN statements to verify that the script executed to that point. Then, place a WRITE and PAUSE inside the IF..THEN to see if it gets past the IF correctly. It would also be helpful to have the WRITE statement give the value of the variables being tested to make sure that the variables are correct going into the script. For example, **WRITE "This is the login name: %LOGIN\_NAME"**.

You can also use the PAUSE command with the MAP command to determine when and where a certain mapping was created in a login script.

- ◆ Make a test login script file either on the local drive or a network drive using any DOS text editor.

Test this script file by using the /S parameter at the DOS prompt:

```
LOGIN /S drive:\filename servername/username
```

If this login script works correctly, copy each statement from this login script into the login script that doesn't work correctly. This way, problems in login scripts can be dealt with by elimination.

# Turning Off the MAPROOT Function in Windows NT/2000/XP

In Windows NT/2000/XP, all drive mappings created using NetWare LOGIN are root mapped. Because of this, programs cannot access directories above the directory that the drive is mapped to. If necessary, you can turn off the default by adding SET MAPROOTOFF="1" as the first line in the login script. This will globally force all NT/2000 workstations using the login script to not map root drives.

## Other Common Login Script Problems

### Losing Path Statements after Login

After each log in, the user would lose a part of the path specified in the autoexec.bat file. The user had PATH statement in autoexec.bat and then had mapped some search drives in the system login script (MAP S1:= MAP S2:= ). PATH statements are set up as search mappings when logged in to the network starting with s1, s2, etc. When you specifically MAP (S1:=), then you overwrite the existing search mapping (PATH). Solution: Doing a MAP INS on the search drives will push the existing statements down the line without overwriting. Using MAP INS S16:= will append the map statements to the end of the line without overwriting those at the beginning.

### Cannot Use the LOGIN and NLIST Utilities

Don't delete the login directory (sys:login) created during network installation. It contains the Novell LOGIN and NLIST utilities. If users run command line utilities, they can use these utilities to log in and view a list of available NetWare servers.

### Cannot Execute External Command

This error occurs with external commands (such as CAPTURE) trying to run inside a login script. To troubleshoot this problem, make sure the COMSPEC is set. Then, make sure login.exe is being swapped out correctly.

### LOGIN.EXE Problems

Because the language-enabled Login.exe is so large, it must be swapped out before large executables can run. capture.exe is a good example of an external program frequently called in a login script. By default, login.exe will try to swap to extended memory first, then to expanded memory, and then to the current drive. If writing to a drive, be sure you have at least the Read and Write rights. If you are using a diskless workstation, this task can be accomplish by using the SWAP command right before the external command. SWAP out to a place on the server you have already mapped to and have been given the appropriate rights (for example, SWAP F:\USERS%\LOGIN\_NAME). This will allow login.exe to be swapped out and give the external program memory to execute. Also, when running an external command, you must either provide a explicit path to the file or have a search drive mapped to the location of the file before running the command. Typically, this will be sys:\public.

### EXIT Command Doesn't Work

Make sure the syntax is right. EXIT "*path:command*" with the quotes is the right syntax. Ensure that the path and command you put in the quotes are not more than 14 characters long. The early

versions of login.exe in NetWare 3.1x and the login.exe in NetWare 4.1 don't support more than 14 characters in the path. Also, remember that EXIT will exit you completely out of the login script, not allowing any other login scripts to run. If there is a user or profile login script that a user needs to execute, you should not use EXIT to get out of the system/container login script.

## **Login Script Does Unexpected Things**

Make sure there are no viruses on the computer. Viruses have caused numerous strange problems with login scripts that have previously executed or should now execute correctly.

Also, temporarily delete any TSRs, as some have been known to cause login problems.

Using an ATTACH command to create an attachment to another NetWare 4.1x server in the same tree can also cause unexpected results such as intermittent premature termination of the login script. The ATTACH command makes a bindery connection which causes any previous eDirectory™ connections to that server to be destroyed. Using CAPTURE with the Server parameter will also cause a bindery attachment to occur.

