

Login Scripts

Disclaimer

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 Novell 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.

Trademarks

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

A complete list of trademarks and their respective owners appears in "Trademarks" on page 97.

Copyright © 1994-1998, 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 Nos. 5,157,663; 5,349,642; and 5,455,932. U.S. Patent Application No. 5,572,528. U.S. and International Patents Pending.

**Novell, Inc.
122 East 1700 South
Provo, UT 84606
U.S.A.**

Login Scripts

Contents

Login Scripts	vii
Understanding	1
Overview	1
The Login Process	1
Related Topics.....	2
The Login Directory.....	2
Login-Related Commands	3
Login Restrictions.....	3
Types of Login Scripts.....	4
Objects That Contain Login Scripts	6
Login Script Conventions	6
Commands and Variables.....	7
Default Login Script.....	8
Sample Container Login Script.....	9
Sample Profile Login Script.....	11
Sample User Login Script	12
Planning	15
Which Login Scripts to Create	15
Where Login Scripts Should Be Located.....	16
Whether to Use Login Scripts from Other NetWare Versions.....	17
Which Commands to Include.....	18
Which Identifier Variables to Use.....	18
Related Topics.....	18
Setting Up	21
Creating a Login Script from a Sample.....	21
Creating Login Scripts.....	22
Copying Login Scripts	22
Printing Login Scripts	23

Using NetWare Administrator to Create Login Scripts	24
Related Topics	25
Using Login Script Commands	26
Using Variables	27
Login Parameters with %n Variables.....	32
Examples of How to Use Identifier Variables	32
Conventions	34
Login Script Commands.....	34

Managing.....39

Modifying Login Scripts	39
Adding SUPERVISOR Bindery Login Capabilities (DOS and OS/2 only)	39

Troubleshooting41

Login Script Conversion Issues for OS/2 Workstations	41
Login Script Edit Problems.....	41
Login Script Execution	44
Login Script Syntax	46
Getting around the One Profile Login Script Restriction	47

Reference49

# 49	
@ (Microsoft Windows*Only).....	51
Attach	52
BREAK	52
CLS (DOS and OS/2 only)	53
COMSPEC (DOS Only)	53
CONTEXT	54
DISPLAY	55
DOS BREAK	56
DOS SET	57
DOS VERIFY (DOS and Windows 3.x only)	57
DRIVE.....	57
ELSE	58
END.....	59
EXIT.....	59
FDISPLAY	60
FIRE PHASERS.....	61
GOTO	63
IF...THEN	64
INCLUDE.....	68

LASTLOGINTIME	70
LOGOUT	70
MACHINE (DOS and Windows 3.1x Only)	71
MAP	71
NOSWAP (DOS only)	74
NO_DEFAULT	74
PAUSE	75
PCCOMPATIBLE	75
PROFILE	76
REM.....	76
REMARK.....	77
SCRIPT_SERVER	78
SET.....	78
SET_TIME	80
SHIFT	81
SWAP (DOS only).....	83
TEMP SET	84
TERM	84
TREE	85
WRITE	86

Examples..... 89

#SEND /A=N	89
@GRPWISE5.....	89
COMSPEC = S2:COMMAND.COM.....	89
Login Scripts for NetWare Clients Contents	89
FIRE 2	91
FIRE 3 RIFLE.WAV.....	91
IF "%1"="ADMIN" THEN MAP *1:SYS:SYSTEM	91
IF DAY_OF_WEEK="FRIDAY" THEN.....	91
IF MEMBER OF "MANAGERS" THEN.....	91
IF PLATFORM = "WIN31" THEN.....	91
IF OS2 THEN	91
Login.....	92
MAP	92
MAP *1:=SYS:.....	92
MAP *1:=%HOME_DIRECTORY	92
MAP *3:=VOL1:PROJECTS\REPORTS	92
MAP *5:=VOL1:ACCOUNTS\NEW.....	93
MAP *6:=VOL1:ACCOUNTS\RECORDS.....	93
MAP *7:=VOL1:MARY\PROJECTS\RESEARCH	93
MAP *8:=VOL1:FORMS.....	93

MAP DISPLAY OFF	93
MAP DISPLAY ON.....	93
MAP ERRORS OFF.....	93
MAP INS *2:=SYS:USERS\%LOGIN_NAME\WIN31.....	93
MAP INS S16:=SYS:APPS\WINAPPS\WIN31.....	94
MAP INS S16:=SYS:EMAIL.....	94
MAP INS S16:=VOL1:APPL\DB.....	94
MAP INS S16:=VOL1:APPL\LOTUS.....	94
MAP INS S1:=SYS:PUBLIC.....	94
MAP INS S2:=SYS:PUBLIC\%MACHINE\%OS\%OS_VERSION.....	94
MAP M:=SHARED.....	95
MAP N:=MODEMS.....	95
MAP O:=SYS:DOC.....	95
MAP P:=SYS:PUBLIC.....	95
PCCOMPATIBLE.....	95
SET PROMPT = "\$P\$G".....	95
SET TEMP = "P:\USERS\%LOGIN_NAME\WIN31\TEMP".....	96
SET USER="mrichard".....	96
SET WP="/u-mjr/b-5".....	96
WRITE.....	96
WRITE "Good %GREETING_TIME, %FULL_NAME.".....	96
WRITE "Weekly progress report is due today.".....	96
WRITE "Your password expires in %PASSWORD_EXPIRES days.".....	96

Trademarks 97

Novell Trademarks.....	97
Third-Party Trademarks.....	97

Index 99

L*ogin Scripts*

Login scripts can simplify network administration once you understand what they have to offer, how they work, and when best to use them. This section provides information on how to create login scripts, when and where to create them, and other relevant information. It also provides detailed information about each login script command, and help with troubleshooting login scripts.

Understanding

In order to maximize the effectiveness of login scripts, it helps to understand the login process, what types of login scripts are available for you to use, and where you can put those login scripts. For you to create effective login scripts, you must also understand the conventions you must follow when creating login scripts. This section provides you with this information.

Overview

Login scripts automatically set up users' workstation environments whenever they log in to the network. Login scripts are similar to configurable batch files and are executed by the LOGIN utility.

You can use login scripts to

- ◆ Map drives and search drives to directories
- ◆ Display messages
- ◆ Set environment variables
- ◆ Execute programs or menus

Login scripts work the same way for Microsoft** Windows**, DOS, and OS/2** workstations.

Some login script commands that apply to Microsoft Windows workstations might not apply to DOS and OS/2 workstations. These differences are indicated in the description of each command.

The Login Process

When a user logs in, the operating system looks for security rights; the user is then asked for a password.

All security information is placed into the NetWare* server's connection list and the user is said to be logged in.

At this point, LOGIN executes one or more login scripts (which initialize environment variables, map network drives, etc.). It is the LOGIN utility that executes the appropriate login scripts.

Four types of login scripts are available, and they can be used separately or together to tailor a custom environment for your users. All four types of login scripts are optional.

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.

Related Topics

Attach

Login Directory

Login Restrictions

LOGOUT

The Login Directory

The Login directory is the sys:login directory created during network installation that contains the 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.

For NetWare users running OS/2**, the corresponding Login directory is sys:login/os2.

Don't delete the Login directory.

Login-Related Commands

Choose from any of the following login-related commands to view information about the command, including its command format and usage information and examples.

- ◆ LOGOUT
- ◆ MAP

Login Restrictions

Login restrictions are limitations on user accounts that control access to the network, such as the following:

- ◆ Requiring a password

If you require a password, you can specify its minimum length, whether it must be changed (and how often), whether it must be unique, and whether the user can change it.

- ◆ Setting number of logins with expired password

You can also specify the number of times a user can log in using an expired password and the number of incorrect login attempts allowed.

- ◆ Setting account limits

If you install Accounting, you can assign account limits (like an account balance or expiration date).

- ◆ Limiting disk space

You can limit the amount of disk space for each user by specifying the maximum blocks available for each user on a volume.

- ◆ Specifying the number of connections

You can limit the number of times a user can log in simultaneously. You can also specify, by node address, which workstations users can log in on.

- ◆ Setting time restrictions

You can restrict the hours during which users can log in. You can assign all users the same hours, or you can restrict users individually.

When a user violates login restrictions, NetWare* disables the account and no one can log in using that username. This prevents unauthorized users from logging in.

The following are some restrictions which apply to login scripts:

- ◆ To use the login script from an Organization, Organizational Unit, or Profile object, users must have the Browse right to the object and the Read right to the object's Login Script property.
- ◆ If users are logged in to a server running NetWare 2 or NetWare 3* and that server is in an NDS* tree, do not create or edit a login script using the SYSCON utility. If you do, the changes to that login script will not appear in the NDS login script. This is because an NDS login script is a property of the User object, while a bindery-based login script is a file in the MAIL directory.

Types of Login Scripts

Login scripts are files containing commands that set up users' workstation environments whenever they log in. Login scripts are similar to batch files and are executed by the LOGIN utility.

There are four types of login scripts: Container, Profile, User, and Default. You can create or modify only the first three types. The Default login script is part of the LOGIN utility, and while you can prevent it from executing, you cannot modify its content.

Container Login Script

A container login script sets the general environments for all users in that container. The LOGIN utility executes container login scripts first. A user can use only one container login script.

A container login script replaces the system login script from NetWare* 3*.

[Click here for more information and to view a sample container login script.](#)

Profile Login Script

A profile login script sets environments for several users at the same time. The LOGIN utility executes a profile login script after the container login script.

A user can be assigned only one profile login script but can then specify other profile login scripts. To specify other profile login scripts, the user can use the Profile list box on the Script tab of the Novell* LOGIN utility. For more information, see Getting around the One Profile Login Script Restriction.

Note: By using the /p option, DOS and OS/2** users can change the profile script.

[Click here for more information and to view a sample profile login script.](#)

User Login Script

A user login script sets environments (such as printing options or a username for electronic mail) specific to a single user. The LOGIN utility executes the user login script after any container and profile login scripts have executed.

A user can have only one user login script.

[Click here for more information and to view a sample user login script.](#)

Default Login Script

The default login script is precoded into the LOGIN utility and cannot be edited. It executes if a user doesn't have his or her own user login script, even if a container or profile login script exists.

The default login script is executed for all users (including user ADMIN) unless you create a user login script--or if you prevent the default login script from executing by including the NO_DEFAULT command in the container or profile login scripts

The default login script contains only essential commands, such as drive mappings to the NetWare utilities.

[Click here for more information and to view the default login script.](#)

Objects That Contain Login Scripts

Login scripts are properties of objects. Only four types of objects can contain login scripts, and each object can contain only a certain type of script:

- ◆ Organization object -- Container login script
- ◆ Organizational Unit object -- Container login script
- ◆ Profile object -- Profile login script
- ◆ User objects -- User login script

Login Script Conventions

This table explains the conventions which must be followed when creating login scripts.

Subject	Convention
Minimum login script	No minimum. All four types of login scripts are optional. Login scripts can have only one line or they can have many. There are no required commands for login scripts.
Case	Either uppercase or lowercase is accepted. Exception: identifier variables enclosed in quotation marks and preceded by a percent sign (%) must be uppercase.
Characters per line	512 characters per line is maximum 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 only 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. WRITE command output displays better if WRITE is repeated at the beginning of each wrapped line.
Sequence of commands	Generally, enter commands in the order you want them to execute, with the following restrictions: ATTACH commands (NetWare 3* only) must precede related MAP commands to avoid prompting the user for a username/password during login. If you use "#" (or @) to execute an external program, it 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.
Blank lines	Blank lines don't affect login script execution. Use them to visually separate groups of commands.
Remarks (REMARK, REM, asterisks, and semicolons)	Lines beginning with REMARK, REM, an asterisk, or a semicolon are comments, which don't display when the login script executes. Use remarks to record the purpose of each command or group of commands.
Identifier variables	Type identifier variables exactly as shown. For the value of an identifier variable to be displayed on the workstation's screen as part of a WRITE command, you must enclose the identifier in quotation marks and precede it by a percent sign (%).

Commands and Variables

Syntax conventions for login script commands are the same as those for workstation text utilities, with one exception: some login script commands must be preceded by the # symbol.

The following example is the syntax for the MAP login script command:

```
MAP [option] drive:=drive:|path <Enter>
```

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 LOGIN utility. Instead, you can create container, profile, or user login scripts.

The following sample of the default login script lists its contents and lets you find out more information about each command in the default login script.

Using This Sample Login Script

Use this sample login script in two ways:

- ◆ As an example of login script commands which you might need to duplicate elsewhere (depending on how you choose to set up and use login scripts within your network)
- ◆ As a way to learn the purpose of various login script commands (click each login script command for an explanation of its purpose and use within the default login script)

Reviewing This Sample Login Script

For a description of the function of any of the specific login script commands contained in this example default login script, click that specific login script command.

For more detailed information about each type of login script command (such as its command format, use, and examples), click login script commands .

```
MAP DISPLAY OFF
```

```
MAP ERRORS OFF
```

```
MAP *1:=SYS:
```

```
MAP *1:=SYS:%LOGIN_NAME
```



```
IF "%1"="ADMIN"THEN MAP *1:SYS:SYSTEM
MAP P:=SYS:PUBLIC
MAP INS S1:=SYS:PUBLIC
MAP INS S2:=SYS:PUBLIC\%MACHINE\%OS\%OS_VERSION
MAP DISPLAY ON
MAP
```

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.

Using This Sample Login Script

You can use this sample login script in two ways:

- ◆ As a starting point for creating your own container login scripts (click [here](#) for instructions)
- ◆ As a way to learn the purpose of various login script commands (click each individual login script command for an explanation of its purpose and use within this sample login script)

Reviewing This Sample Login Script

For a description of the function of any of the specific login script commands contained in this sample login script, click that specific command.

For more detailed information about each type of login script command (such as its command format, use, and examples), click this [login script commands link](#).

```
MAP DISPLAY OFF
```

```
MAP ERRORS OFF
```

```
MAP *1:=SYS:
MAP *1:=%HOME_DIRECTORY.
IF "%1"="ADMIN"THEN MAP *1:SYS:SYSTEM
IF OS2 THEN
MAP P:=SYS:PUBLIC
ELSE
MAP INS S1:=SYS:PUBLIC
MAP INS S2:=SYS:PUBLIC\%MACHINE%\OS\%OS_VERSION
END
IF PLATFORM = "WIN31" THEN
MAP INS *2:=SYS:USERS\%LOGIN_NAME\WIN31
MAP INS S16:=SYS:APPS\WINAPPS\WIN31
SET TEMP = "P:\USERS\%LOGIN_NAME\WIN31\TEMP"
END
MAP INS S16:=VOL1:APPL\LOTUS
MAP INS S16:=SYS:EMAIL
MAP M:=SHARED
MAP N:=MODEMS
MAP O=SYS:DOC
IF MEMBER OF "MANAGERS" THEN
MAP *3:=VOL1:PROJECTS\REPORTS
END
```

```
COMSPEC = S2:COMMAND.COM
```

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

```
MAP DISPLAY ON
```

```
MAP
```

```
WRITE
```

```
WRITE "Good %GREETING_TIME, %FULL_NAME."
```

```
WRITE "Your password expires in %PASSWORD_EXPIRES days."
```

```
FIRE 3 RIFLE.WAV
```

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 sample 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.

Using This Sample Login Script

You can use this sample login script in two ways:

- ◆ As a starting point for creating your own container login scripts (click [here](#) for instructions)
- ◆ As a way to learn the purpose of various login script commands (click each individual login script command for an explanation of its purpose and use within this sample login script)

Reviewing This Sample Login Script

For a description of the function of any of the specific login script commands contained in this sample login script, click that specific login script command.

For more detailed information about each type of login script command (such as its command format, use, and examples), click this login script commands link.

```
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

PCCOMPATIBLE
```

Sample User Login Script

The following sample 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.

Using This Sample Login Script

You can use this sample login script in two ways:

- ◆ As a starting point for creating your own container login scripts (click here for instructions)
- ◆ As a way to learn the purpose of various login script commands (click each individual login script command for an explanation of its purpose and use within this sample login script)

Reviewing This Sample Login Script

For a description of the function of any of the specific login script commands contained in this sample login script, click that specific login script command.

For more detailed information about each type of login script command (such as its command format, use, and examples), click this login script commands link.

```
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 troubleshooting team
```

```
SET WP="/u-mjr/b-5"
```

```
SET USER="mrichard"
```

```
#SEND /A=N
```

```
PCCOMPATIBLE
```


Planning

Planning which login scripts you will create and where to put those login scripts before you begin creating them can save you a great deal of time and frustration. This section helps you with that planning. In addition, it helps you to understand whether you should use login scripts created for earlier versions of NetWare, and what commands and variables you might want to put in your login scripts when you get ready to create them.

Which Login Scripts to Create

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

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. These 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.

Since up to three login scripts can execute whenever a user logs in, conflicts can occur. If this happens, the last login script to execute (usually the user login script) overrides any conflicting commands in a previous login script.

Choose from the following to view an example (with line-item descriptions) of each type of login script:

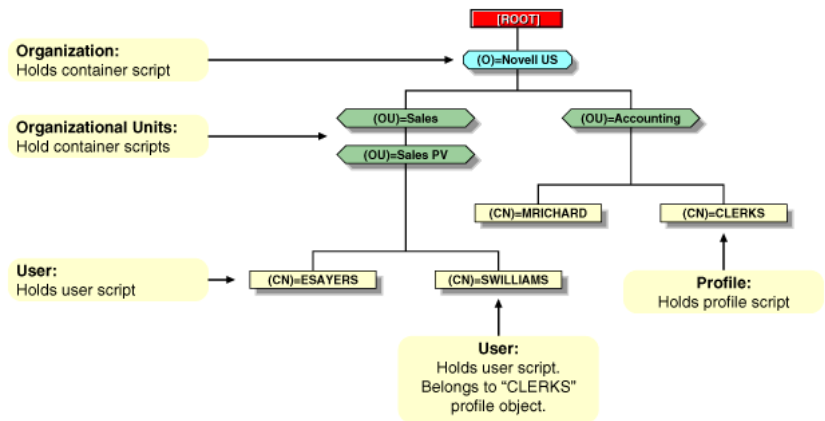
- ◆ Container login script
- ◆ Profile login script

- ◆ User login script
- ◆ Default 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 NDS* tree and how they affect users.



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

- ◆ For user ESAYERS: The SALES PV's container login script executes first, followed by ESAYERS's user login script.
- ◆ For user SWILLIAMS: The SALES PV's container login script executes first, followed by the default login script.
- ◆ For 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 figure, although there are two levels of container objects above users ESAYERS and SWILLIAMS, only the container login script 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.

Since 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 the one profile login script restriction.

Whether to Use Login Scripts from Other NetWare Versions

When you use `install.nlm` to upgrade versions of NetWare* with NDS*, the login scripts in the former Login directory are automatically added to the NDS database as properties of their respective objects.

In most cases, these earlier login scripts are compatible with versions of NetWare running NDS. However, we recommend you review all your login scripts to see if they should be updated to take advantage of NDS functionality.

In versions of NetWare prior to NetWare 4*, login scripts for OS/2** and DOS were separate. In subsequent versions, only one login script is used for all environments. You should note that there are some differences when the login scripts for other versions of NetWare were used for OS/2 versus DOS workstations.

Which Commands to Include

You can use login scripts to automatically set up your users' workstation environments. To accomplish this, several login script commands are commonly used.

These commands accomplish such tasks as making or breaking network connections, establishing access to network resources, setting up workstation environments, displaying text, and tracking a user's last login time.

Reviewing the following topics might be useful before deciding which commands to include in login scripts:

- ◆ Types of login scripts
- ◆ Login script conventions
- ◆ Which login scripts to create
- ◆ Which identifier variables to use

To view a list of available login script commands by the type of task they perform, and to subsequently obtain details about each command, see [Login Script Commands](#).

Which Identifier Variables to Use

With many login script commands, you can take advantage of identifier variables to make your login script more efficient and flexible.

Identifier variables allow you 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 login script command applicable to many users instead of limiting it to one user.

Related Topics

[Conventions](#)

Examples of how to use identifier variables

Login parameters with % *n* variables

Setting Up

You can create login scripts from scratch, or by following the special instructions provided in this section, you can copy a sample login script from this help file into one or more login scripts you create. You can then modify the copied sample login script to better fit the needs of your network users. This copy-and-paste option helps reduce the number of syntax errors which sometimes occur when creating login scripts, thus reducing the time it takes to create and troubleshoot login scripts.

Creating a Login Script from a Sample

To create a login script from any of the samples included in this help file, complete the following steps.

1. Open the help window showing the sample login script.
2. Make a copy of the sample login script by highlighting the text and copying it to the Clipboard buffer.
3. (Optional) Save the script to a new file using an editor like Notepad by pasting it in and then saving the file.
4. Open the empty login script you want to create using NetWare* Administrator.
5. Paste the text you copied from the sample login script into the new login script.
6. If you want to make any changes to the sample login script, make them now.
7. Click OK to save the new login script.

Creating Login Scripts

To create (or modify) login scripts and to copy one object's login script into another's, you can use NetWare* Administrator.

The main difference in creating container, profile, and user login scripts is the object you select to contain the login scripts:

- ◆ 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.
- ◆ User login scripts are assigned to User objects.

All four types of login scripts use the same conventions, commands, and variables.

Copying Login Scripts

You can create login scripts by copying them from another login script. You can also create login scripts using the NetWare* Administrator utility.

Use the following instructions to copy all or part of a login script and paste it into another object's login script.

Prerequisites

- ◆ A workstation running NetWare Administrator
- ◆ The Write property right to the object that will contain the login script.
- ◆ The objects whose login scripts you will be working with (Organization, Organizational Unit, Profile, or User object) must already exist.

Procedure

1. Launch the NetWare Administrator utility.
2. Select the object whose login script you want to copy.

For information about moving around in the browser and selecting objects, click Help from the menu bar.

3. Click Object > Details > Login Script.
4. In the login script text box, highlight the text you want to copy.
5. Press Ctrl+Insert to copy the highlighted text.

The highlighted text has been placed in clipboard memory and can be pasted into another login script.

6. To save the login script and close the Details dialog box, click OK.
7. Using the browser, select the object whose login script you want to paste the copied text into.
8. Click Object > Details > Login Script.
9. In the login script text box, place the cursor where you want the copied text to appear.
10. Press Shift+Insert to paste the copied text into the login script.
11. To save the login script and close the Details dialog box, click OK.

Printing Login Scripts

Use the following instructions to print a login script.

Prerequisites

- ◆ A workstation running DOS 3.30 or later
- ◆ Read and File Scan property rights to the object to be printed

Procedures

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.

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
```

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

Using NetWare Administrator to Create Login Scripts

Purpose

Use the following instructions to create any of the three user-created types of login scripts (container, profile, or user).

Prerequisites

- ◆ A workstation running NetWare* Administrator
- ◆ The Write property right to the object that will contain the login script
- ◆ The Organization, Organizational Unit, Profile, or User object that you plan to assign the login script to must already exist

Procedure

1. Click Netware Administrator.
2. Using the browser, select the object whose login script you are creating or modifying.
3. Click Object > Details > Login Script.
4. Enter the login script commands and information into the login script text box.

For a description of all login script commands, see [Login Script Commands or Using Login Script Commands](#). For login script examples, see [Types of Login Scripts](#).

5. Click OK to save the login script and then close the Details dialog box.

If the login script you just created was a container or user login script, you're finished.

If the login script you just created was for a Profile object, continue with Step 6.

6. (Profile login scripts only) Using the browser, select the User object that needs to use the profile login script.

7. Click Object > Details > Login Script.

8. Enter the name of the Profile object in the Default Profile field located under the login script text box.

9. To save the Profile object name and close the Details dialog box, click OK.

Now you must add the User object as a trustee of the Profile Object.

10. Using the browser, select the Profile object.

11. Click Object > Trustees of This Object > Add Trustee.

12. Enter the name of the User object that is using this Profile object.

13. Make sure the Browse object right and the Read property right are checked, and 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.

Related Topics

[Types of login scripts](#)

[Objects that contain login scripts](#)

[Using variables](#)

Using Login Script Commands

From this page, you can link to any login script command for more information. You can also link to a list of login script commands arranged by the type of functions you want the command to perform.

#

@

ATTACH

BREAK

CLS

COMSPEC

CONTEXT

DISPLAY

DOS BREAK

DOS SET

DOS VERIFY

DRIVE

EXIT

FDISPLAY

FIRE PHASERS

GOTO

IFTHEN

INCLUDE

LASTLOGINTIME

MACHINE

MAP

NO DEFAULT

NOSWAP

PAUSE

PCCOMPATIBLE

PROFILE

REMARK

SCRIPT SERVER

SET

SET TIME

SHIFT

SWAP

TEMP SET

TERM

TREE

WRITE

Using Variables

The following lists each of the available login script identifier variables.

For more information on a specific login script command, [click here](#).

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 (94, 95, 96, etc.)
	YEAR	All four digits of year (1994, 1995, 1996, etc.)
Time	AM_PM	Day or night (am or pm)
	GREETING_TIME	Time of day (morning, afternoon, or evening)
	HOUR	Hour (12-hour scale; 1 through 12)
	HOUR24	Hour (24-hour scale; 00 through 23; 00=midnight)
	MINUTE	Minute (00 through 59)
User	SECOND	Second (00 through 59)
	%CN	User's full login name as it exists in NDS*
	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 NDS and bindery-based NetWare*. Spaces are replaced with underscores.
	LAST_NAME	User's last name (surname) in NDS, or full login name in bindery-based NetWare
	LOGIN_CONTEXT	Context where user exists
	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
Network	FILE_SERVER NETWORK_ADDRESS	NetWare server name IPX* external network number of the cabling system (8 digit hexadecimal number)
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 (MSDOS, OS2, etc.)
	OS_VERSION	Operating system version on the workstation (3.30, etc.)
	P_STATION	Workstation's node number (12-digit hexadecimal)
	PLATFORM	Workstation's operating system platform: DOS, OS2, WIN (Windows** 3.1), WNT (Windows NT*), or W95 (Windows 95*)
	SHELL_TYPE	Version of the workstation's DOS shell (1.02, etc.); supports NetWare 2 and 3 shells and NetWare 4* Requester for DOS.
	SMACHINE	Short machine name (IBM**, etc.)
	STATION	Workstation's connection number
	WINVER	Version of the workstation's Windows operating system
NetWare Mobile	DIALUP	0 = Not using dialup 1 = Using NetWare Mobile* Client dialup profile
	OFFLINE	0 = Not offline 1 = Disconnected login

DOS environment	<variable>	Any DOS environment variable can be used in angle brackets (<path>, etc.). To use a DOS environment variable in MAP, COMSPEC, and FIRE PHASERS commands, add a percent sign (%) in front of the variable. For example: MAP S16:=%<path>
Miscellaneous	ACCESS_SERVER	Shows whether the access server is functional (TRUE = functional, FALSE = not functional)
	ERROR_LEVEL	An error number (0 = no errors)
	%n	Replaced by parameters the user enters at the command line with the LOGIN utility
Object properties	property name	You can use property values of NDS 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

Login Parameters with %n Variables

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

When a user logs in, he or she can enter additional parameters that the LOGIN utility passes to the login script. The login script 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 executing the LOGIN utility (using either File > Run or Start > Run, or the command line).

The %0 variable is replaced by the name of the NetWare* server the user entered at the command line, and %1 is replaced by the full (not shortened to eight characters) user's login name. The remaining variables change, depending on what the user types when executing LOGIN. The %n variables must precede all command line options.

The SHIFT login script command allows you to change the order in which these %n variables are substituted.

Examples of How to Use Identifier Variables

Example 1: Using the %LAST_NAME variable

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

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

```
Hello, SMITH
```

Similarly, when Mary Jones logs in, the message she sees is

```
Hello, JONES
```

In these examples, the user's actual last name was substituted for the LAST_NAME variable in the command upon login.

Example 2: Using the %GREETING_TIME variable

If user 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"
```

Example 3: Using a DOS Environment variable

To use environment variables as identifiers, enclose them in angle brackets. The following example uses the DOS environment variable "path:"

```
WRITE "my path is "; <path>
```

The text displayed on the screen is similar to

```
my path is z.:;y.:;c:\windows
```

Example 4: Using % n variable in WRITE statements

The % n variable can be used in WRITE statements if included within the quotes:

```
WRITE "My login name is %1."
```

Suppose a login script contains the following commands:

```
IF "%2"="SALES" THEN
```

```
WRITE "Meeting today"
```

```
END
```

```
IF "%3"="LEGAL" THEN
```

```
WRITE "Report is due tomorrow"
```

```
END
```

If user Ron logged in by typing

```
LOGIN COUNT\RON SALES MARKETING
```

then the login script would substitute the values Ron entered at the keyboard for the % n variables in the login script, as shown here:

%0=COUNT

%1=RON

%2=SALES

%3=MARKETING

Since %2 is replaced by SALES, the message "Meeting today" is displayed on Ron's screen. However, since %3 is replaced by MARKETING, Ron doesn't see "Report is due tomorrow."

Conventions

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

- ◆ Identifier variables are used most often with commands such as IF...THEN, MAP, and WRITE. They can also be used with commands you can specify a path for, such as COMSPEC.
- ◆ Type the variable exactly as shown.
- ◆ To use environment variables as identifiers, enclose them in angle brackets.
- ◆ 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.)

[Click here for examples.](#)

Login Script Commands

Login scripts are files containing commands that set up users' workstation environments whenever they log in. Login scripts are similar to batch files in that they cause a series of commands to be run. DOS batch files run DOS commands, while login script files run login script commands; DOS batch files are executed by DOS, while login scripts are executed by the LOGIN utility.

Some login script commands apply only in specific environments, such as when the workstation is an OS/2** workstation, while other login script commands apply regardless of the workstation environment.

For detailed information about each available login script command, see the following lists. These list is designed to help you identify which specific login script commands to use depending on what you need your login scripts to accomplish.

Network Connection and Access to Resources

ATTACH

CONTEXT

DRIVE

MAP

TREE

Login Script Execution

BREAK

FIRE PHASERS

GOTO

IFTHEN

INCLUDE

NO_DEFAULT

PAUSE

PROFILE

SCRIPT SERVER

SHIFT

SWAP

Workstation Environment

COMSPEC

DOS BREAK

DOS SET

MACHINE

NOSWAP

PCCOMPATIBLE

SET

SET TIME

TEMP SET

Text File Usage

@

#

CLS

DISPLAY

DOS VERIFY

EXIT

FDISPLAY

TERM

WRITE

Other

LASTLOGINTIME

REMARK

Managing

When it becomes necessary to modify your existing login scripts, this section provides the information you need to accomplish that task. In addition, this section shows you how to add SUPERVISOR bindery login capabilities.

Modifying Login Scripts

To create or modify login scripts and to copy one object's login script into another's, you can use NetWare* Administrator . Modifying login scripts is much the same as creating login scripts. Therefore, refer to the instructions for creating login scripts when you want to modify a login script.

Adding SUPERVISOR Bindery Login Capabilities (DOS and OS/2 only)

The SUPERVISOR bindery login is an administrative login that is not represented by an NDS* object. Even an NDS User object called SUPERVISOR is not the same as the SUPERVISOR bindery login. You can log in as the bindery SUPERVISOR using the LOGIN /B option. The bindery SUPERVISOR is kept with each server and is not affected by replication of NDS objects.

If the nds.vlm program is not loaded when you login, LOGIN automatically performs a bindery login, thus the /B option is not required except from an OS/2** workstation.

Troubleshooting

If you are having trouble creating, editing, or running login scripts, this section can help you. It explains the syntax associated with login script commands, (errors in which are a common cause of login script problems). Ever see an error message telling you the login script could not be edited? Help to solve that problem can be found in this section as well.

Login Script Conversion Issues for OS/2 Workstations

If a NetWare* 3.11 user, for example, has both an OS/2** and a DOS login script, both of those login scripts are combined into a single login script when that user is upgraded to NetWare 4* or later.

The commands from the OS/2 login script appear first in the upgraded login script, followed by the commands from the DOS script.

When the user logs in, the LOGIN utility determines whether the user is using OS/2. Because of IF statements put in the script during migration/upgrade, LOGIN executes only the section of the upgraded script that contains the corresponding commands. For an example of a related login script command, see the PCCOMPATIBLE and EXIT commands in the sample profile login script.

Login Script Edit Problems

Occasionally, you might have problems when editing a login script. Here are some of the potential problems, information about the cause of each problem, and guidelines for correcting or working around the problems.

Locating Other Login Script Troubleshooting Information

For more information on login scripts, you can also connect through the Internet to <http://support.novell.com>. Once there, select Knowledge Base

to open the search page, and then either enter an exact TID number (such as 1203151) or enter the following into the search field:

troubleshooting AND "login scripts"

You can also select Forums instead of Knowledge Base to connect you to the Forums page. Once there, you must register, if you have not already done so, and then log in. Once you are logged in, you can search the support forums for any topic including troubleshooting login scripts, or post questions to be answered by Novell's SysOps.

Make one of the following World Wide Web connections for more information:

- ◆ Americas: <http://support.novell.com>
- ◆ EMEA: <http://support.novell.de>
- ◆ Asia Pacific: <http://support.novell.com.au>

Getting an Error When Trying to Modify Login Scripts

When editing a login script, you might get an error message indicating that the login script file cannot be opened or cannot be saved. The following are possible causes for this situation, as well as suggested solutions.

Cause: A user's workstation locks up while they are logging in. The login script becomes locked open.

Solution: Use `monitor.nlm` to find and release the connection which is locked. (Look for a connection titled NOT-LOGGED-IN.)

Cause: The user's login script contains a `#` command used to run a program which is external to the login script, such as a user menu or a word processing application. The login script remains open until that external program is closed. In the case of a login script launching an application that the user might hold open all day, the login script will also be held open all day. Login scripts cannot be edited when they are held open.

Solution: Replace the `#` command with the `@` command in the login script. The `@` command does not hold the login script open when it calls an external command.

Cause: A Container, User, or Profile login script can become locked open when the user is executing the login script and the script pauses for user input (for example, using the PAUSE command in the script). If the user does not enter any input when prompted, the login script cannot be edited because it is "locked open." NetWare* Administrator generally returns the error code "NWDS-4.0-624: The login script file could not be opened. Return Code: 1729:128."

Solution: You can work around this problem by doing one of the following:

- ◆ After getting the error message in NetWare Administrator, complete the following steps.
 1. Open the login script again and make changes to it.
 2. Before clicking OK, make a copy of the login script by highlighting the text and copying it to the clipboard buffer (Ctrl+C).
 3. (Optional) Save the script to a new file using an editor like Notepad (use Ctrl+V to paste the text into the editor).
 4. Delete the entire login script in NetWare Administrator and click OK. The changes will be saved.
 5. Open the now empty login script in NetWare Administrator and paste (Ctrl+V) the text back into the login script.
 6. Click OK. The login script will be saved, and you will not see the error message this time.
- ◆ If you know the server you are logged into and which server you are reading the NDS* objects from, or you are in a single-server environment, complete the following steps:
 1. Locate the user that is holding the login script file open by loading MONITOR and looking at each connection to find the user that has a file open in the sys:netware directory.
 2. Either have the user log off temporarily or clear the user's connection to close the login script.
 3. Now open the login script, edit it, save it, and let the user log back in.

- ◆ If you are in a multiserver environment and the NDS tree has been partitioned with several servers holding replicas of the partition in which this Container, Profile, or User object's login script is located, complete the following steps.
 1. Determine which server is returning the error.
 2. Locate another server that has a replica of the same partition and log in to that server.
 3. Run NetWare Administrator from the server you logged into in step 2.
 4. Now edit the login script from this server and click OK to save it. The modified login script will be replicated to the other servers holding replicas of the affected partition.
- ◆ Various problems related to editing login scripts might occur when editing the login scripts using NETADMIN or NWADMIN95. Edit login scripts instead with NWADMIN32.

Login Script Execution

If your login scripts do not execute the way you expect them to, there are several possible reasons for the problem.

- ◆ The first, and often most common, reason for incorrect login script execution is syntax error.

Check the troublesome commands in your login script for syntax errors. Also be sure to check the syntax of the command which executes just before the one you are having trouble getting to execute properly. For information on proper login script syntax, see [Login Script Syntax](#).

- ◆ If you create a User login script but do not put the minimum required commands into it, such as a drive mapping to the sys:public directory, your user might be unable to perform simple network tasks such as mapping a drive. That is because once you create a user login script, the default login script no longer executes; the user login script executes instead.

For information on user and default login scripts, see [Types of Login Scripts](#).

- ◆ Using variables in a login script is a good way to provide a less generic login interface for users. Improper use of variables can cause unexpected results; so ensure you have made proper use of variables in your login script.

For more information on login script variables, see [Using Variables](#).

- ◆ Although you cannot technically assign more than one profile login script to a user, there might be times when you need to have multiple profile login scripts execute for a single user.

For more information on this topic, see [Getting around the One Profile Login Script limitation](#).

- ◆ Your users might need access to a NetWare* bindery-based server.

You can enable a bindery-based login to that server for them. For more information see [Whether to Use Login Scripts from other NetWare Versions](#) or [Adding SUPERVISOR Bindery Login Capabilities](#).

- ◆ Sometimes the reason a login script does not execute as you intended it to is due to the login script's location in the Directory tree.

See [Where Login Scripts Should Be Located](#).

- ◆ If the login script command you chose simply does not seem to be doing what you expected it to do, perhaps you just need to choose a different login script command.

For information about login script commands, see [by related task](#).

- ◆ Login scripts execute in the following order: Container, Profile, User (if there is one), and then Default (which does not execute if a User login script or the command `NO_DEFAULT` is contained in another login script). If some of the drive mappings you intended for a user are being changed when the user logs in, it might be due to the order in which the login scripts are being executed.

Check each login script to determine whether the execution of one is overriding the commands in another. For more information see [Types of Login Scripts](#).

Login Script Syntax

If you are having problems getting one or more commands in your login scripts to run properly, verify that the commands follow the proper syntax. Syntax errors are a common cause of improper login script execution.

Syntax conventions for login script commands are the same as those for workstation text utilities, with one exception: Some login script commands must be preceded by the # (or @) symbol.

Following is an example of the syntax for the MAP login script command:

```
MAP [option] drive:=drive:|path <Enter>
```

The command syntax is described below:

MAP Words in uppercase letters are keywords that must be included in the command and spelled exactly as shown. However, it doesn't matter if you type them in uppercase or lowercase letters.

[] Square brackets indicate that the enclosed item is optional.

| A vertical bar means you can use either the item to the left of the bar or the item to the right, but not both. In the MAP example above, you can enter either the drive letter or a complete path.

drive Words in italics are variables. Replace variables with information specific to your task.

option Options or parameters for each command are listed with the command. Options and parameters can often be abbreviated.

<Enter> Angle brackets indicate a key you should press.

[[]] Nested square brackets indicate that all enclosed items are optional. However, if you use the items within the innermost brackets, you must also use the items within the outer brackets.

Getting around the One Profile Login Script Restriction

Users can belong to only one profile login script so, as a rule, only one profile login script can be executed for any user. There are ways to cause the equivalent of different or multiple profile login scripts to run. However, you can execute a different profile login script from the one which normally executes for a user by specifying other profile login scripts.

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

```
LOGIN username /p profile_object
```

You can also specify a different profile login script using NetWare* Administrator. On the user's Login Script page, browse until you find the profile login script you want to associate with the user, select it, and then click OK. For more information on associating a profile login script with a user, see the NetWare Administrator help.

You also can assign users to more than one Group object. Then use the MEMBER OF group identifier variable to specify that different parts of a login script execute, depending on the Group objects that the user belongs to. To accomplish this, use the IFTHEN command prompt and MEMBER OF group identifier variable.

To view an example of the IF...THEN command used with the MEMBER OF group identifier variable, see the sample container login script.

Reference

In this section, you can look up login script commands by the command name.

#

Use this command (the # symbol) 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. You can use this command for either a DOS or a Windows* login.

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. If a login script contains a # command used to run a program which is external to the login script, such as a user menu or a word processing application, the login script remains open until that external program is closed. In the case of a login script launching an application the user might hold open all day, the login script will also be held open all day. Login scripts cannot be edited when they are held 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 *path* 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.

Using #

If you want the LOGIN utility to execute a program that is external to the login script and then continue with the login script, enter the # command followed by the name of the file you want to execute.

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

LOGIN swaps to extended or expanded memory or to disk unless NOSWAP is specified on the command line or in the login script.

NOSWAP prevents LOGIN from being swapped out of conventional memory. Then, if the station does not have enough memory to handle both LOGIN and the # command, the # command fails but the rest of the login script executes normally.

For more information, see SWAP (DOS Only).

Example

If you are running the Novell* Client* for Windows** NT** or the Novell Client for Windows 95**, you can use the Automatic Client Update (ACU) method of installation to update workstation client software without having to physically visit each workstation and manually update the software.

This method of client software update relies on the use of a file called unattend.txt, which can be called from the login script. To call this file from the login script for a workstation running Novell Client for NT, 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
```

@ (Microsoft Windows*Only)

Use this command (the @ symbol) when using a Microsoft** Windows** workstation to execute a program that is external to the login script. You cannot use this command with a DOS login.

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. If a login script contains a # command used to run a program which is external to the login script, such as a user menu or a word processing application, the login script remains open until that external program is closed. In the case of a login script launching an application the user might hold open all day, the login script will also be held open all day. Login scripts cannot be edited when they are held open.

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.

Using @

If you want the LOGIN utility 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 (symbol) followed by the name of the file you want to execute.

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

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 utility 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 GroupWise5 (GRPWISE5.EXE) is located in a search drive.

If you do not have a search drive mapped to the directory where the external 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.

In networks running NetWare 3* or earlier versions, users connect to multiple file servers using the ATTACH command. In versions of NetWare with NDS*, users no longer need to attach separately to multiple servers.

When users log in to the NDS tree, they automatically have access to any resources in the tree to which they have rights. Rights to resources are verified through authentication.

BREAK

Use BREAK ON to allow the user to terminate execution of the login script. The default is BREAK OFF.

Command Format

```
BREAK ON|OFF
```

Using BREAK

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.

Including BREAK ON in a login script does not affect the DOS Ctrl+Break check. For more details, see DOS BREAK.

When the BREAK option is ON, type-ahead keyboard input is not saved in the buffer.

CLS (DOS and OS/2 only)

Use CLS to clear the display from the workstation's screen during the login process.

Command Format

CLS

Using CLS

When a user logs in, a login script might display messages on the workstation screen.

If the CLS command is added to the login script, any messages generated by commands earlier in the login script are cleared from the screen.

COMSPEC (DOS Only)

To execute the DOS command processor from the network, use COMSPEC in the login script. Specify the directory which the DOS command processor (command.com) is to be loaded from.

COMSPEC is originally set when DOS is booted. It can be reset after you log in to change the location that command.com loads from while you're in the network.

COMSPEC should only be used if the command processor is loaded at bootup from a floppy disk, or if it is not available after the user logs in to the network. If the DOS files are on the network, you can still have COMSPEC set the directory path to a network drive so that the command processor can be loaded from that network drive.

Users running virtual DOS sessions under OS/2** should not use this command in the login script. For more information, see Novell's web site at <http://www.novell.com>. (Click Technical Support > Knowledgebase, and then select OS/2-related information from the list of products.)

Command Format

COMSPEC=[*path*] command.com

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

Using COMSPEC

If users are running DOS from a network directory, first map a search drive in the login script to that directory and then add the COMSPEC command to the login script.

You might want to map a fake root to the DOS directory. For information about mapping a fake root, see MAP.

If all users use the same version of DOS from the network, you can add the COMSPEC command to the container login script.

If more than one version of DOS is available on your network, a network directory should exist for each DOS version. In this case, you can put COMSPEC commands in either profile or user login scripts, to make sure each workstation accesses the version of DOS it needs.

If users are running DOS from their local drives, do not add COMSPEC to login scripts.

To use an environment variable as the value in a COMSPEC command, precede it with the percent sign (%), as follows:

COMSPEC=% *environment variable*

CONTEXT

Use CONTEXT to set a user's current context in the NDS* tree.

Command Format

CONTEXT *context*

Using CONTEXT

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

Similarly to the workstation CX utility, you can enter a complete name to move down through the context, or you can use periods to move up toward the root of the tree.

CONTEXT does not support all options that the CX workstation utility does. It only sets the context.

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 on a workstation's screen when the user logs in.

This command works best with an ASCII file. If you use DISPLAY with a word processing file, printer and word processing codes are displayed with the text.

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 you want to display.

Using DISPLAY

When you use DISPLAY to display the contents of a file on the screen, the exact characters in the file, including any printer and word processing codes, appear on the workstation screen. (To display only the text and suppress codes, use FDISPLAY.)

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

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 Mondays. Add the following lines to the container login script:

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

DOS BREAK

Use DOS BREAK to set the Ctrl+Break checking level for DOS.

If DOS BREAK is set to ON, you can terminate a program (other than the login script) by pressing Ctrl+Break. The default is DOS BREAK OFF.

This command does not apply to OS/2** workstations.

This command is different from the BREAK command that terminates a login script. For more details, see BREAK.

Command Format

```
DOS BREAK [ON|OFF]
```


Using DOS BREAK

Enter the following command in the login script:

DOS BREAK ON

DOS SET

See SET for information about the DOS SET command.

DOS VERIFY (DOS and Windows 3.x only)

Use DOS VERIFY to verify that data written to a local drive is not written to a bad sector and can be read without an error.

Command Format

DOS VERIFY [ON|OFF]

Using DOS VERIFY

The DOS COPY command does not automatically verify that data copied to a local drive can be read after the copy.

To assure verification of each copy operation after login, add the DOS VERIFY ON command to the login script.

Another option is to use the /V option at the command line with each COPY operation.

The default in the login script is DOS VERIFY OFF.

This command might not work with some software that is copy protected.

DRIVE

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

On OS/2** workstations, DRIVE is effective only during execution of the login script. When the login script completes, you're returned to the drive that was in effect when the LOGIN command was issued.

Command Format

DRIVE [*drive* [* *n*]

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

Using DRIVE

Unless this command is in your login script, the default drive is set to the first network drive, which is often assigned to your home directory when you log in.

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.

Example

Suppose you expect to work on only one project for several days and the files for that project are located on drive S:. You can use the DRIVE command to set your default drive to S: so you won't have to change your default drive manually every time you log in.

First, make sure you've mapped drive S: to the correct directory in your login script. Then enter the following command in the login script:

```
DRIVE S:
```

ELSE

An optional statement used only with the IF...THEN command. It specifies that the commands which follow it are executed only when the condition identified between the IF and THEN portions of the command is true. If used, the ELSE statement must be on a separate line.

END

Identifies the previous command as the last command in an IF...THEN command statement. (Must be used only when there is more than one line in an IF...THEN command statement.)

EXIT

Use EXIT to terminate execution of the login script and execute an external program.

This command does not apply to OS/2** workstations.

Command Format

```
EXIT ["filename [parameters]"]
```

Using EXIT

The length of information between quotes is not limited in Microsoft** Windows**. However, the length of information between quotes is limited in DOS; the length of information between quotes can't exceed your keyboard buffer length minus 1 (commonly 15 - 1 = 14 characters).

You can use the EXIT command in a login script to stop the login script and execute a program, such as a word processing or menu program.

You can also use EXIT in an IF...THEN statement, so that the login script stops and exits to an external program 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 EXIT command and continues executing.

Because EXIT stops the login script, make sure you put this command either at the end of the login script or at a point within the script where you intend execution to stop.

If the program you are executing with the EXIT command requires any DOS paths or NetWare* search drives to be set, make sure they are specified in the login script ahead of the EXIT command.

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

The EXIT with a command works only on IBM** compatible workstations running DOS. Therefore, if your DOS workstation has a machine name different from IBM_PC specified in its net.cfg file, you must add the PCCOMPATIBLE login script command to the login script.

For more information about the PCCOMPATIBLE command, see PCCOMPATIBLE (DOS Only).

To run the equivalent of the EXIT command on a Microsoft Windows workstation, you must use two commands: @ <command> followed on the next line by EXIT.

Examples

Suppose the workstation's long machine name is IBM_PC. To execute a batch program called TRAINING.BAT when the login script is finished, add the following line at the end of the login script:

```
EXIT "TRAINING"
```

If you are using a Hewlett Packard** computer and you have changed the long machine name to HE_PAC in the net.cfg file, add the following lines at the end of the login script:

```
PCCOMPATIBLE
```

```
EXIT "TRAINING"
```

Suppose you want the login script to exit to a word processing program when the user logs in on Mondays but not on other days. You could add the following IF...THEN statement to the login script:

```
IF DAY_OF_WEEK="MONDAY" THEN EXIT "WP"
```

FDISPLAY

Use FDISPLAY to show the text of a word processing file on a workstation's screen 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.

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 you want to display.

Using FDISPLAY

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

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

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 Mondays.

Add the following lines to the container login script:

```
IF DAY_OF_WEEK="Monday" THEN
```

```
FDISPLAY SYS:PUBLIC\MESSAGES\SYSNEWS.TXT
```

```
END
```

FIRE PHASERS

FIRE PHASERS signals the workstation to emit a phaser sound. This phaser sound is the result of the phasers.wav sound file being executed.

Command Format (Windows Only)*

`FIRE n sound file`

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

Replace *sound file* with the name of the sound file you want to play when this command is executed. You can use any .wav or platform-compatible sound file. A .wav sound file is generally used. (One or more .wav files can usually be found in the windows/system directory.)

Command Format (DOS Only)

FIRE *n*

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

Using FIRE PHASERS

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

Example

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

FIRE 4

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

FIRE %*environment variable*

Example (Windows Only)*

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

FIRE 3 rifle.wav

To use an environment variable as the number of times to fire, use % before the variable, followed by the name of the sound file, as follows:

FIRE %*environment variable* rifle.wav

For more information about using identifier variables, see [Which Identifier Variables to Use and Using Variables.](#)

GOTO

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

Command Format

GOTO *label*

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

Using GOTO

Set BREAK ON in your login script before experimenting with GOTO loops so that you can break out of a login script if necessary.

For more information about the BREAK login script command, see BREAK

Do not use GOTO to enter or exit a nested IFTHEN statement. This usage confuses the program.

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 when you want the login script to perform an action only under certain conditions.

Command Format

```
IF conditional [AND|OR [conditional]] THEN
```

```
commands
```

```
[ELSE
```

```
command]
```

```
[END]
```

Replace *conditional* with identifier variables. For more information about identifier variables, see Using Variables.

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

Using IF...THEN

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, 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)

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
```

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
```

The following line is a compound conditional that means "If it is 11:59:59 p.m.":

```
IF HOUR24="23" AND MINUTE="59" AND SECOND="59"
```

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.

```
IF DAY_OF_WEEK="TUESDAY" THEN
```

```
WRITE "Staff meeting today at 10 a.m."
```

```
INCLUDE SYS:PUBLIC\UPDATE
```

```
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 subscripts can be text files that contain valid login script commands (any of the commands explained in this section) or login scripts that belong to a different object you have rights to.

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.

Using INCLUDE

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

You can create and edit text file subscripts using any text editor. Subscripts do not have 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 subscripts.

`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 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.

Examples

To execute a text file called `script.new` (located in the `Vol1:` volume) as a subscript, add the following line to your main login script:

```
include vol1:admin\users\script.new
```

Now 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`.

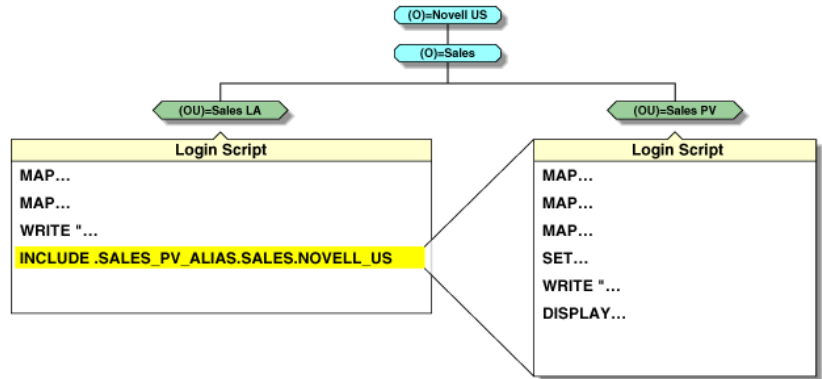
You've decided that the login scripts for the two different groups of users are very similar. In fact, you decide that the `SALES_LA` users could use the same login script as the `SALES_PV` users, but with a few more drive mappings.

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.

Command Format

LASTLOGINTIME

Using LASTLOGINTIME

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

LOGOUT

A procedure that breaks the network connection and deletes drives mapped to the network (unless they are permanent).

If you log out without specifying a NetWare* server name in the LOGOUT command, the station connections and drives mapped to all servers are terminated.

To log out from one server and remain attached to the other servers, specify the server name in the LOGOUT command.

Make sure at least one of the remaining drives is mapped to the Public directory of a NetWare server that you are still attached to. Otherwise, you can't access NetWare utilities.

MACHINE (DOS and Windows 3.1x Only)

Use MACHINE to set the DOS machine name (such as IBM** or EDIT_ROOM) of the workstation. The MACHINE command is necessary for some programs (such as NetBIOS) written to run under PC-DOS.

The MACHINE command does not apply to OS/2** workstations.

Important: Do not confuse the MACHINE command with the identifier of the same name. The identifier is used with a preceding percent sign (%) in MAP and WRITE statements. The identifier reads its value from the net.cfg file.

It is unlikely that you will need to use this command. However, it is provided in case you come upon a program that requires it.

Command Format

MACHINE=name

Using MACHINE

The machine name can be up to 15 characters. (Longer machine names are truncated to 15 characters.)

For example, to specify that the machine name is IBM_PS2, add the following line to the login script:

```
MACHINE=IBM_PS2
```

MAP

Use MAP to map drives and search drives to network directories.

Command Format

```
MAP [options] [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, or a Directory Map object.

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 NDS* 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 NDS server that isn't the current server, begin the path with the server's name.

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.
- ◆ **INS:** Inserts a drive mapping between existing search mappings.
- ◆ **DEL:** Deletes a drive mapping, making that drive letter available for other mapping assignments.
- ◆ **ROOT:** Maps a fake root. Windows** NT** and OS/2** are always mapped to the root. Some applications require their executable files to be located in a root directory.

Since 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 native environment forces a map root on all drives. To prevent a forced map root in a Windows NT 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.

It is possible to have a Volume object name that conflicts with a physical volume name. (For example, object ACCT is an Accounting volume, but there is also an ACCT which is a physical volume.)

Therefore, if you prefer to map a drive to the physical volume name, use MAP P.

- ◆ N (Next): When used without specifying a drive number or letter, maps the next available drive.

Using MAP

If you use MAP to automate drive map assignments during execution of the login script, users don't have to manually 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. In a Windows environment, however, do not map over the local drive containing the Windows directory.

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

NOSWAP (DOS only)

Use NOSWAP to prevent the LOGIN utility from being moved out of conventional memory into higher memory (if available) or onto the disk to execute a # command and LOGIN at the same time.

Command Format

NOSWAP

Using NOSWAP

LOGIN always swaps to extended or expanded memory unless NOSWAP is specified on the command line or in the login script.

If you do not want LOGIN to be temporarily stored in higher memory or on the workstation's disk, use the NOSWAP command. NOSWAP prevents LOGIN from being swapped out of conventional memory.

Then, if the workstation does not have enough memory to handle both LOGIN and the # command, the # command fails but the rest of the login script executes as usual.

If you want LOGIN to be swapped out of conventional memory immediately every time a # command is executed, place the SWAP command in the login script, before the # command.

NO_DEFAULT

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

Command Format

NO_DEFAULT

Using NO_DEFAULT

If you do not want to create any user login scripts, and you do not want the default user login script to run, add this command to either the container or the profile login script.

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

PAUSE

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

Command Format

PAUSE

Using PAUSE

Enter this command in the login script at any point you want a pause to occur.

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. The LOGIN utility then waits for a key to be pressed before it executes the rest of the login script.

PCCOMPATIBLE

Use PCCOMPATIBLE to enable the EXIT "command" login script command to work if the workstation's LONG MACHINE NAME is not IBM_PC.

This command does not apply to OS/2** workstations.

Command Format

PCCOMPATIBLE

Using PCCOMPATIBLE

If the computer is an IBM** PC-compatible machine and not an IBM PC, use PCCOMPATIBLE in the login script to inform the LOGIN utility that the machine's long name is something other than IBM_PC. The LONG MACHINE NAME (COMPAQ**, or others) must be included in the net.cfg file.

Place the PCCOMPATIBLE command anywhere before EXIT in the login script

Example

If you have a Hewlett Packard** computer and have changed the LONG MACHINE NAME to HE_PAC in the net.cfg file, put the following commands in your login script:

```
PCCOMPATIBLE
```

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
```

REM

The REM command identifies all text which follows on the same line as a remark. Remarks are not displayed on the user's screen. Remarks can be several lines long, but each line must begin with the REM command.

The remark (REM) command is often used to remind the person who created the login script of the purpose or function of one or more other login script commands which follow the remark command.

REMARK

Use REMARK (or REM) to insert explanatory text into the login script.

Command Format

REM[ARK] [*text*]

or

* [*text*]

or

;*text*]

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

Using REMARK

To include explanatory text in the login script, begin a line with REMARK, REM, an asterisk (*), or a semicolon (;).

Any text that follows these symbols is ignored when the LOGIN utility executes the login script. Remarks do not appear on the screen.

Using remarks can make the script much easier for you or other users to read and understand.

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 (REMARK, REM, an asterisk, or a semicolon).

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 from where the bindery login script is read.

`SCRIPT_SERVER` has no effect on NetWare 4* and later users.

Command Format

`SCRIPT_SERVER server_name`

Using SCRIPT_SERVER

This command has no effect on NDS*.

SET

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

For OS/2** workstations, `SET` commands affect the environment only while the login script is running; the settings disappear when `LOGIN` terminates.

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.

Variables set in the login script for an OS/2 workstation affect the environment only while the login script is running. Therefore, all variables are considered TEMP with OS/2 workstations.

Using SET

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

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

SET commands do not have to be included in login scripts. For example, you might decide that it is easier to put 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>).

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) character. See your DOS or OS/2** manual for more information.

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.

Command Format

```
SET_TIME ON|OFF
```

Using SET TIME

The default value is based on the Novell* Client* for Windows** 95** SET STATION TIME advanced property setting SET_TIME ON. If you include SET_TIME OFF in the login script, the workstation time does not update to the server's time.

SHIFT

Use SHIFT to change the order in which %n identifier variables are interpreted in the login script. SHIFT allows users to enter LOGIN parameters in any order.

Command Format

SHIFT [n]

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

Using SHIFT

You can shift up to 10 login script variables (command line arguments).

When users execute 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 left.

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 Accts.

Mary also has a command in her login script to map a drive to her Lotus 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
```

```
IF "%2"="ACCNTS" THEN
```

```
MAP G:=SYS:ACCNTS
```

```
IF "%2"="LOTUS" THEN
```

```
MAP S16:=SYS:APPL\LOTUS
```

```
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 right 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 right so that %2 now becomes WP.

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

SWAP (DOS only)

Use SWAP to move the LOGIN utility out of conventional memory into higher memory (if available) or onto the disk. This allows execution of a # command and LOGIN at the same time.

Command Format

SWAP [*path*]

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

Using SWAP

By default, the LOGIN utility always swaps to extended or expanded memory, unless NOSWAP is specified on the command line or in the login script.

Important:

The SWAP option doesn't work with the DR DOS** 6.0 EMM386 Memory Manager option unless upper memory is disabled.

If you specify a path in the SWAP command, LOGIN swaps into the directory you specified. If the directory specified in that path does not exist or if you do not have rights there, LOGIN prompts you for another path.

If you don't specify a path, LOGIN swaps either into higher memory (if available) or to the current drive. If LOGIN tries to swap to the current drive and you don't have rights to that drive, LOGIN prompts you for a path to use. If you specify a valid path, LOGIN always swaps to the specified path.

Then, if the workstation doesn't have enough memory to handle both LOGIN and the # command, the # command fails but the rest of the login script executes as usual.

TEMP SET

For information on the TEMP SET command, see the SET command.

TERM

This command is normally used only for Novell* Application Launcher* scripts, a component of Z.E.N.works*. This command does not apply to OS/2** workstations.

Command Format

TERM *nnn*

Using TERM

Replace *nnn* with an error level. Any error level between 000 and 999 is valid.

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.

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.

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 NDS* tree attachments.

Use TREE to attach to another NDS tree within your network and to access its resources.

The TREE command changes the "focus" of your login script so that all NDS object references in subsequent script commands (for drive mappings, print captures, etc.) apply to the NDS 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]]
```

Using Tree

Replace *tree_name* with the name of the NDS tree that you want the user to attach to.

Replace *complete_name* with the user's complete name (Distinguished Name) for the NDS tree that the user is attaching to. The complete name establishes your 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 NDS tree named CORP, add the following line to her 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.

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 Variables for a complete list of variables.

Using WRITE

Text 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 exactly 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	Makes a carriage return occur.
\n	Starts a new line of the text.
\"	Displays a quotation mark on the screen.
\7	Makes a beep sound.

In addition to the semicolon, you can use additional 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, modulus
+ -	Add, subtract

>>	Shift left or right (1000 >> 3 becomes 1)
<<	
\7	Makes a beep sound.

Examples

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

```
WRITE "Hello"
```

To display the user's last name (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.

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"
```


Examples

#SEND /A=N

Executes the external SEND utility so that messages will not be received from the console or from other users.

@GRPWISE5

Executes grpwise5.exe, which launches the GroupWise* program on the user's workstation.

COMSPEC = S2:COMMAND.COM

Sets COMSPEC to the DOS command processor, located in the DOS directory (in the second search drive).

Login Scripts for NetWare Clients Contents

Understanding

Overview

The Process of Logging in

The Login Directory

Login Restrictions

Types of Login Scripts

Objects That Contain Login Scripts

Login Script Conventions

Planning

Which Login Scripts to Create

Where Login Scripts Should Be Located

Whether to Use Login Scripts from Other NetWare Versions

Which Commands to Include

Which Identifier Variables to Use

Setting Up

Creating Login Scripts

Creating a Login Script from a Sample

Copying Login Scripts

Printing Login Scripts

Using Variables

Using Login Script Commands

Managing

Modifying Login Scripts

Adding SUPERVISOR Bindery Login Capabilities

Troubleshooting

Login Script Syntax

Login Script Execution

Login Script Edit Problems

Reference

Login Scripts by Related Task

FIRE 2

Causes the phaser sound to occur twice when the user logs in.

FIRE 3 RIFLE.WAV

Makes the rifle sound occur three times, to tell the user that the login process is complete.

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

If the login name is ADMIN, it maps the first drive to sys:system instead of to the user's home directory.

IF DAY_OF_WEEK="FRIDAY" THEN

Determines whether the current week day is a Friday, and executes subsequent command if it is.

IF MEMBER OF "MANAGERS" THEN

If the user who logs in is a member of the MANAGERS Group object, the next command (MAP *3:=VOL1:PROJECTS\REPORTS, in this case) is executed.

IF PLATFORM = "WIN31" THEN

If the user who logs in uses the WIN31 platform, the next command (MAP INS *2:=SYS:USERS\%LOGIN_NAME\WIN31, in this case) and any subsequent commands are executed.

IF OS2 THEN

If the user logs in from an OS/2** workstation, the system is to perform the next login script command (MAP P:=SYS:PUBLIC, in this case).

Login

The procedure that provides access to the network by using the LOGIN utility.

MAP

Displays a list of all drive mappings on the user's screen. The MAP DISPLAY ON command must precede the MAP command if the MAP DISPLAY command was previously set to OFF.

MAP *1:=SYS:

Maps the first drive to volume Sys:.

MAP *1:=%HOME_DIRECTORY

Maps the first drive to the user's home directory. If the user has no home directory set, the following alternate command can be used.

```
IF "%HOME_DIRECTORY" = "" THEN  
  
MAP *1:=SYS:USER/%LOGIN_NAME  
  
ELSE  
  
MAP *1:=%HOME_DIRECTORY  
  
END
```

MAP *3:=VOL1:PROJECTS\REPORTS

Maps the third network drive to the Reports directory. Because this command is part of the IF MEMBER OF "MANAGERS" THEN command statement, it will execute only for users who are members of the Managers Group object.

MAP *5:=VOL1:ACCOUNTS\NEW

Maps the fifth network drive (after those assigned in the container login script) to the New subdirectory.

MAP *6:=VOL1:ACCOUNTS\RECORDS

Maps the sixth network drive (after those assigned in the container login script) to the Records subdirectory.

MAP *7:=VOL1:MARY\PROJECTS\RESEARCH

Maps Mary's seventh network drive (after those assigned in the container and profile login scripts) to the Research subdirectory in her home directory.

MAP *8:=VOL1:FORMS

Maps Mary's eighth network drive (after those assigned in the container and profile login scripts) to the Forms directory.

MAP DISPLAY OFF

Prevents MAP commands from displaying on the screen.

MAP DISPLAY ON

Allows MAP commands to display on screen.

MAP ERRORS OFF

Prevents mapping errors from displaying on the screen.

MAP INS *2:=SYS:USERS\%LOGIN_NAME\WIN31

Maps the next available drive to the user's Microsoft** Windows** directory as specified by the %LOGIN_NAME identifier variable.

MAP INS S16:=SYS:APPS\WINAPPS\WIN31

Maps the next available search drive to the Microsoft** Windows** directory (sys:apps\winapps\win31) for the named group. (In this case, the group is Win31 as identified in the IF MEMBER OF "WIN31" THEN command statement.)

MAP INS S16:=SYS:EMAIL

Maps the next available search drive to the e-mail directory.

MAP INS S16:=VOL1:APPL\DB

Maps the first available search drive (after those assigned in the container login script) to the directory that contains the database program.

MAP INS S16:=VOL1:APPL\LOTUS

Maps the next available search drive to the directory that contains Lotus**.

MAP INS S1:=SYS:PUBLIC

Maps the first search drive to the sys:public directory where NetWare* utilities are stored. (Make this command part of an IF...THEN command instead of its own separate command only when there are non-DOS workstations on the network.)

If the user logs in from an OS/2** workstation, this drive mapping is not included in the default login script.

MAP INS S2:=SYS:PUBLIC\%MACHINE%\%OS%\%OS_VERSION

Maps the second search drive to the directory where DOS is stored. (Make this command part of an IF...THEN command instead of its own separate command only when there are non-DOS workstations on the network.)

If the user logs in from an OS/2** workstation, this drive mapping is not included in the default login script.

MAP M:=SHARED

Maps the M: network drive letter to the Directory Map object called Shared. This drive mapping assumes the User object for the user who is logging in to the network is located in the same context as the Shared Directory Map object.

MAP N:=MODEMS

Maps network drive N: to an Alias object called Modems. This Alias object is located in the same container as this user's User object. The Modems Alias object, however, points to a Directory Map object which represents the Modems directory located in another container.

MAP O=SYS:DOC

Maps drive O: to a directory called Doc.

MAP P:=SYS:PUBLIC

If the user logs in from an OS/2** workstation, maps drive P: to sys:public.

If the user is not using an OS/2 workstation, this drive mapping is not included in the default login script.

PCCOMPATIBLE

Allows the EXIT command to function when a workstation does not have the machine name set to IBM_PC.

SET PROMPT = "\$P\$G"

Sets the prompt to display the user's current directory path, followed by the > symbol.

SET TEMP = "P:\USERS\%LOGIN_NAME\WIN31\TEMP"

Using drive letter P: sets the Microsoft** Windows** Temp directory to a subdirectory of the user's Microsoft Windows directory (users\%login_name\win31).

SET USER="mrichard"

Sets Mary's username (MRICHARD) for the electronic mail program.

SET WP="/u-mjr/b-5"

Set's Mary's environment variables for her word processing application.

WRITE

Displays a blank line between the list of mappings and subsequent lines.

WRITE "Good %GREETING_TIME, %FULL_NAME."

Displays a greeting to the user. For example: "Good morning, MARY JONES."

WRITE "Weekly progress report is due today."

Writes the following message to the user's screen when the user logs in: "Weekly progress report is due today."

**WRITE "Your password expires in
%PASSWORD_EXPIRES days."**

Displays a message indicating the number of days before the user's password expires. Forxample: "Your password expires in 10 days."

Trademarks

Novell Trademarks

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

Internetwork Packet Exchange and IPX are trademarks of Novell, Inc.

NDS is a trademark of Novell, Inc.

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

NetWare 3 is a trademark of Novell, Inc.

NetWare 4 is a trademark of Novell, Inc.

NetWare 5 is a trademark of Novell, Inc.

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 Application Launcher is a trademark of Novell, Inc.

Novell Client is a trademark of Novell, Inc.

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

Third-Party Trademarks

Compaq is a registered trademark of Compaq Computer Corporation.

DR DOS is a registered trademark of Caldera, Inc. in the United States and other countries.

Hewlett Packard is a registered trademark of Hewlett-Packard Company.

IBM is a registered trademark of International Business Machines Corporation.

Lotus is a registered trademark of Lotus Development Corporation.

Microsoft is a registered trademark of Microsoft Corporation.

OS/2 is a registered trademark of International Business Machines Corporation.

Windows is a registered trademark of Microsoft Corporation.

Windows 95 is a trademark of Microsoft Corporation.

Windows NT is a registered trademark of Microsoft Corporation.