

**QUICK for NDS**  
**eDirectory Command Line Tool**

Quick for NDS v1.5x



**Verlag Wolfgang Schreiber-Neumann**

<http://www.WolfgangSchreiber.de>





Copyright 1999-2006 by Verlag Wolfgang Schreiber-Neumann  
<http://www.WolfgangSchreiber.de>

Printed: 17-Oct-06

All rights reserved. No part of this manual or software may be reproduced or modified without the prior written permission of the publisher.

Though all functions and descriptions of this software have been produced and tested carefully by the author, errors can never be completely excluded. Author and publisher cannot be made liable for any damage caused by incorrect usage, incomplete documentation, or software malfunction.

The publisher reserves the right to make changes to any part of the software or the documentation without obligation to notify any person or entity.

Publisher and author gratefully accept suggestions for future revisions of this product.









# Table of Contents

<b>1</b>	<b>OVERVIEW .....</b>	<b>12</b>
1.1	WHY QUICK ? .....	12
1.2	QUICK AND THE DIRECTORY .....	12
1.3	LIMITATIONS AND COMPATIBILITY .....	13
1.4	RISKS AND PRECAUTIONS .....	13
1.5	NOTES AND TROUBLESHOOTING .....	14
1.6	QUESTIONS AND ANSWERS .....	15
<b>2</b>	<b>SYNTAX, FEATURES, AND INSTALLATION .....</b>	<b>16</b>
2.1	GENERAL SYNTAX .....	16
2.2	AVAILABLE OPTIONS .....	16
2.3	GENERAL OPTIONS .....	16
2.4	NOTATION .....	18
2.5	VARIABLE IDENTIFIERS IN COMMANDS .....	19
2.6	FINDING OBJECTS – THE ‘FIND’ OPTION .....	21
2.7	ERROR LEVELS .....	22
2.8	INSTALLATION.....	22
2.9	VERSION HISTORY, RECENT CHANGES .....	23
<b>3</b>	<b>QUICK FOR NDS: AVAILABLE OPTIONS .....</b>	<b>25</b>
3.1	ACCOUNT: CHECK OBJECT ACCOUNT INFORMATION .....	25
3.2	ALIAS: VIEW OR CREATE ALIAS OBJECTS .....	25
3.3	ATTRIBUTE: VIEW OR CHANGE OBJECT ATTRIBUTES .....	26
3.4	CHECK: INSPECT OBJECT INTEGRITY .....	29
3.5	CLONE: CLONE/COPY EXISTING OBJECTS .....	29
3.6	CONTEXT: GET OR SET THE DEFAULT CONTEXT.....	30
3.7	COPY: CLONE/COPY EXISTING OBJECTS .....	31
3.8	CREATE: CREATE NEW OBJECT .....	32
3.9	DELETE: DELETE OBJECTS .....	33
3.10	DUMP: SHOW ALL AVAILABLE OBJECT ATTRIBUTES .....	33
3.11	EDIRECTORY: RETRIEVE EDIRECTORY INFO.....	34
3.12	EQUIVALENCIES: LIST SECURITY EQUIVALENCIES .....	34
3.13	ERRMSG: CONVERT ERROR NUMBER INTO MESSAGE.....	35
3.14	EXISTS: CHECK OBJECT EXISTENCE.....	36
3.15	FORALL: EXECUTE COMMAND FOR MATCHING OBJECTS .....	36
3.16	FORSET: EXECUTE COMMAND FOR EACH MEMBER OF ATTRIBUTE SET.....	37
3.17	GROUPS: SHOW OR CHANGE GROUP MEMBERSHIPS .....	39
3.18	HOMEDIR: HANDLE HOME DIRECTORIES .....	39
3.19	LIST: LIST OBJECTS .....	40
3.20	LOGINSRIPT: DISPLAY LOGIN SCRIPTS.....	41
3.21	MEMBER: SHOW OR CHANGE GROUP MEMBERS .....	42
3.22	MOVE: MOVE OBJECTS TO A DIFFERENT CONTAINER.....	43
3.23	OPTIONS: LIST AVAILABLE COMMAND LINE OPTIONS.....	44
3.24	PASSWORD: CHANGE OBJECT PASSWORD .....	44
3.25	RENAME: RENAME EXISTING OBJECT .....	45
3.26	RESTRICT: SHOW OR CHANGE OBJECT RESTRICTIONS.....	46
3.27	RIGHTS: GET OBJECT'S RIGHTS .....	47
3.28	SECURITY: RUN EDIRECTORY SECURITY CHECK .....	48
3.29	SERVER: EXECUTE COMMAND ON SERVER.....	48
3.30	SHOW: SHOW OBJECT AND SELECTED PROPERTIES .....	49
3.31	SYNTAX: SHOW ALL QUICK SYNTAXES .....	50
3.32	VISIBLE: CHANGE VISIBILITY OF OBJECTS.....	51
<b>4</b>	<b>LICENSE AGREEMENT .....</b>	<b>54</b>







# 1 Overview

## 1.1 Why QUICK ?

QUICK is an eDirectory utility that can be used for many purposes:

- **QUICK is command line oriented**  
QUICK offers the opportunity to retrieve or change information that usually is accessible by eDirectory menus like ConsoleOne, only. These QUICK functions can also be called from Batch jobs, thereby allowing automated execution for any number of items, or at a scheduled time.
- **QUICK output can be redirected**  
To document an existing network with its security structure, it is necessary that outputs are comprehensive and can be redirected to files, or printers. The QUICK functions '*SHOW*' and '*TOTAL*' will often have to be redirected.
- **QUICK uses unknown eDirectory options**  
QUICK allows you to retrieve information that may not be available using regular eDirectory commands. You may for example extract the schema information, or read the creation time of eDirectory objects.
- **QUICK simplifies system management**  
Many QUICK functions comprise in a single command the features of multiple steps of eDirectory tools. Examples for this simplified management are options like '*SHOW*', '*CLONE*', '*ForAll*', and many more.
- **QUICK allows bulk operations**  
Many QUICK functions allow you to use wildcards in object names, hence operate on a set of matching objects.  
QUICK often allows the '*/Sub*' operator to continue operations through sub-containers.  
Explicitly, the '*ForAll*' '*ForMmb*' commands allow the execution of any command on any set of matching objects.

## 1.2 QUICK and the Directory

QUICK is a utility that directly accesses NDS with eDirectory function calls to retrieve, or change directory information.

eDirectory is a hierarchical database hosted by one or more file servers. eDirectory represents network resources as objects in the eDirectory tree. For example, objects can represent organizations, people, or physical resources such as printers and workstations.

Just like NWAdmin accesses and changes eDirectory, you may use QUICK for NDS to read or edit eDirectory. You cannot circumvent the eDirectory security clearance by using non-Novell tools: NWAdmin or QUICK or any other tool will just allow you to do what the system administrators have allowed you to do.

More detailed information about the eDirectory architecture and interfaces is available in the [programming documentation](#) published by Novell.

## 1.3 Limitations and Compatibility

- **eDirectory versions**

QUICK is compatible with NDS and eDirectory versions v4.x to v8.x running on any platform (NetWare v4.x and above, Linux, Windows, etc).

- **Operating system and client software**

To run QUICK for NDS you will need a 32bit Windows platform (e.g., Win9x, WinNT, or Win2k, WinXP, Win2k3, etc) and the Novell Client installed. The Microsoft client for Novell does not provide a full interface to eDirectory.

- **Memory requirements**

A minimum of 32MB RAM is recommended, though QUICK for NDS may run with even less RAM. However, some operations that generate larger lists in environments with huge numbers of eDirectory objects may require more RAM.

- **Disk space requirements**

Since QUICK for NDS itself does not need disk space except for the executable and the related DLLs, it will typically need less than 2MB of hard disk space and can be installed on a shared drive, e.g., on a NetWare server.

## 1.4 Risks and Precautions

- **Do not use QUICK thoughtlessly!**

QUICK for NDS is a tool that allows very easy access and manipulation of eDirectory. The main objective during the development of QUICK for NDS was to allow calling all functions from the command line or from a batch job without requiring additional user interaction, or confirmation.

QUICK for NDS is a powerful tool, capable of performing major eDirectory modifications. There is a risk of causing damage to your security system when using QUICK for NDS without caution: many eDirectory changes can be reversed, but

before issuing commands like '*QUICK Delete User=\**' take a minute to think about the consequences.

To protect you and your file server from those kinds of problems it is strongly recommended to regularly backup your eDirectory with NDS-aware utilities.

- **Can you break the rules?**

eDirectory offers server based security clearance. The server interprets all commands and requests from clients. eDirectory checks your privileges and rejects unauthorized requests. Whatever tool you use - it is not possible to circumvent this control. QUICK for NDS will not allow you to get more rights than what has been granted to you by eDirectory (i.e., the eDirectory Administrator).

- **Do your actions have to make sense?**

While eDirectory verifies your privileges and rejects unauthorized requests, eDirectory does not make a check for unwise operations. QUICK for NDS allows you to act towards eDirectory as thoughtful or brainless as you wish.

If you have Administrator rights and intend to use QUICK for NDS, you might wish to do so after your first cup of coffee in the morning.

You might want to create an organizational unit in your existing tree (or create a separate tree) to test your actions before operating in the production environment.

## 1.5 Notes and Troubleshooting

- **ZEN objects**

QUICK for NDS works with eDirectory objects. Since some applications (especially NAL or ZENworks) store information outside of eDirectory, QUICK operations may succeed only partially.

To be on the save side, you should test your actions in a test environment before working with your production system.

- **Evaluation copy**

The executable QUICK.EXE is identical for the evaluation copy and the full version. The full version comes with QUICK.INI that contains license information. Make sure this INI file is in the same path as QUICK.EXE

If QUICK.INI is missing or corrupted, QUICK for NDS will run as evaluation copy and display random information messages about this fact.

- **Support or questions**

The fastest way to solve standard problems or questions with QUICK is to post a question with a reference to "QUICK" in the subject on this news forum:

[news://support-forums.novell.com/novell.support.edirectory.netware](https://support-forums.novell.com/novell.support.edirectory.netware)

For problems that go beyond such support or that require engineering, you may contact [wstools@WolfgangSchreiber.de](mailto:wstools@WolfgangSchreiber.de) - a minimum charge of \$50 may apply.

- **Upgrade Protection**

You can upgrade to a newer version of Quick for NDS at a reduced price, if you have purchased a previous version of Quick for NDS:

- Within 1 year of purchase: 80% (!) discount (100% (!) discount for upgrade per download or email)
- Within 2 years of purchase: 60% discount
- Within 3 years of purchase: 40% discount
- Within 4 years of purchase: 20% discount

## 1.6 Questions and Answers

- **NDS versus eDirectory**

**Q:** I have deployed eDirectory on my servers – can I continue to use ‘QUICK for NDS’?

**A:** Basically, “eDirectory” is just a new name for Novell’s Directory Service, NDS. QUICK for NDS is fully compatible with eDirectory and has been tested with the current versions of eDirectory (v4.x to v8.x).

- **Trustee Rights**

**Q:** I would like to be able to assign trustee rights to a folder, is this possible using QUICK?

**A:** You may view and set trustee rights to eDirectory OUs or other objects with the “QUICK ACL” syntax. However, QUICK for NDS does not have an option to set trustee rights, since QUICK operates on the directory, only. Trustee rights are stored in the file system.

You could use another standard Novell command line, RIGHTS.EXE, to grant such rights.

Sometimes it may even be helpful to combine both with commands like:

```
QUICK ForAll [...] "RIGHTS [...]"
```

- **Nag screens**

**Q:** Though I have purchased a full copy of QUICK for NDS, I get random nag screens.

**A:** Make sure that the license information, stored in QUICK.INI, is located in the same path as QUICK.EXE

Unlicensed versions of QUICK are “Trial-Ware”: they share the same EXE and have full functionality, but display random nag messages.

## 2 Syntax, Features, and Installation

### 2.1 General syntax

*QUICK* <option> <parameter list> <general options>

### 2.2 Available options

A detailed overview on these options can be found in chapter 3 (QUICK for NDS: Available Options).

An overview about all QUICK options is available by calling '*QUICK Options*'. A short syntax overview for any given option is available by entering an option without any additional parameters: '*QUICK* <option>' (e.g. '*QUICK Show*').

See the list of available options at the start of chapter 3: QUICK for NDS: Available Options

### 2.3 General Options

Some general options can be combined with nearly all other options (where it makes sense, otherwise the general option will be ignored).

<i>/Container</i> =<container>	Start container for the operation
<i>/Tree</i> =<treename>	Specify a target tree
<i>/Sub</i>	Recurse through sub-containers
<i>/Quiet</i>	Reduce the text output
<i>/Confidence</i> =Low Medium High	Specify confidence level
<i>/Alias</i> =Ref Deref	(De)reference Alias Objects

These general options can be used to select the target tree and container ('/Tree=<name>', '/Cont=<name>'), to specify the display mode ('/Quiet').

- **/Tree**  
Without this option, QUICK for NDS will act on the default tree. If there are multiple trees in your environment, this option allows you to select another than your default tree. Note that QUICK for NDS is licensed per tree and using it on multiple trees requires the appropriate license.
- **/Container**  
This option defines the relative context of object names: if you want to get information on the object "JDoe.Sales.Utah.ACME" you can enter "QUICK Show .JDoe.Sales.Utah.ACME", specifying the absolute name. Alternatively you may enter "QUICK Show JDoe /Cont=.Sales.Utah.ACME", or even "QUICK Show .JDoe.Sales.Utah.ACME /Cont=[Root]". While at first glance this may seem irrelevant, this will have an effect on the display: in the first case, all dependent object names (like groups belonged to) will be displayed as relative names to the current context. In the third case the root will be the relative context and this will make all displayed names be relative to the root i.e., with full context.
- **/Sub**  
Many QUICK for NDS options like 'Clone', 'ForAll', 'Move' allow recursive operations by adding the '/Sub' parameter to specify that the operation affects objects throughout all subordinate contexts.
- **/Quiet**  
If you specify this option, some informational messages will be suppressed, thereby slightly reducing the text output. This might be useful to focus on relevant messages in batch operations with numerous QUICK for NDS calls.
- **/Confidence**  
This option specifies the preferred server target in environments with multiple replicas: The parameters 'Low' and 'Medium' use the default (nearest) replica, while the option 'High' forces QUICK for NDS to send requests to the master replica. (Note: currently the Novell Client does not differentiate between low and medium confidence). The default is "/Confidence=Med".
- **/Alias**  
This option becomes relevant if Alias objects are involved. By default, eDirectory dereferences Alias objects: Assume you have a user object "James Bond" and an alias object "007" that references that user. With a command like "QUICK Show 007" or "QUICK Attr 007 \*", you would receive information about the user object "James Bond" and its attributes. If you'd rather get information about the Alias object itself, add the command option "/Alias=Ref" to your command line, and QUICK will return information about the Alias, rather than the object it references. The default is "/Alias=Deref".

General options can be combined and are typically added at the end of the command line.



## 2.4 Notation

**Abbreviating the option name:** You may abbreviate the option name as long as it stays unique, and as long as the option cannot be confused with any other option.

The option 'Groups' may for example be abbreviated to its first letter (*QUICK G Admin*) since no other option starts with the same letter. The option 'ForAll' may be abbreviated to its first four letters (*QUICK For user=\* ...*) since another option starts with the same three letters.

**Case sensitivity:** The options and parameters are not case sensitive.

**Syntax notation:** Depending on the chosen option, one or more additional parameters must/may be entered: this document uses angle brackets ('<', '>') to identify parameter names (e.g.: <object>). When entering the command, do not type the angle brackets.

Alternative parameters are separated by '|'.

Optional parameters are listed in square brackets ('[', ']').

- **<otype>**

The term <otype> stands for the object type of an eDirectory object; all types recognized by eDirectory are accepted, even those added by your own schema extensions. If the type contains blanks you should enclose it in double quotes.

Therefore, the syntax "QUICK List <otype>" could be applied like:

```
QUICK list User
QUICK list "organizational unit"
```

- **<object>**

The term <object> stands for the name of an eDirectory object (e.g., 'Admin'). Most options, especially those that accept wildcards, will also accept the specification of the object type (e.g., 'User=A\*'). Again, if the name contains blanks you must enclose it in double quotes (e.g., 'User="This User"').

If you are specifying name at the command line, these can be names relative to the current context (e.g., "Admin", or "Admin.X1."), or they can be absolute references starting with a dot (e.g., ".Admin.ISDep.us").

- **Wildcards**

Using wildcards: If any option accepts wildcards, this parameter will be represented as <name\*>. In this case the string may include an asterisk (\*), such as 'A\*' or 'Gr\*'.

The following examples show how to use wildcards for untyped names:

```
c*      Any object whose left-most name begins with a "c" character.
M*y    Any object that begins with "M" and ends with "y" such as Mary.
```

- **More complex methods to find objects**

Instead of using object names and wildcards, you may enter more complex search filters to locate and act on objects. Many QUICK commands allow to replace the "<otype>=<object>" format for selecting objects by a more sophisticated eDirectory search. This allows you to base your object selection on a powerful and individual search.

A basic example of such a search could look like:

```
QUICK show "Find:((Base Class=User)&&(Surname=Sch*))"
```

A more detailed description of this feature and the available syntax can be found in the chapter 'Finding objects – the 'Find' option' (→ p.21).

- **Quotes**

As usual in DOS commands, you need to enclose parameters in double quotes if any of the parameters contains a blank, or some other reserved DOS character like pipes (e.g., the symbols "<", ">", or "|"). If a parameter does not contain any of these characters, you may (but do not have to) enclose it in double quotes.

**Notation sample**

Assuming that the syntax for a command is

```
QUICK Show <object*> [<attr>]
```

Then possible calling formats of this command option include ...

```
QUICK Show someone.oul.org
QUICK Sh User=A* "Common Name"
QUICK Sh ou1.ou2.us *
```

**Output redirection**

Program output can be redirected to files or printers with the standard DOS commands: (e.g. *QUICK Show Admin > c:\x.txt*).

**Iterations and Loops**

While QUICK itself does not provide an option to build internal loops, e.g., to create 100 user objects named "test1" to "test100", you may easily combine QUICK with existing DOS commands. The DOS command "FOR" allows creating such loop iterations – enter "FOR /?" for details.

Creating 10000 users ('Test1' through 'Test10000') could be done with a single command like

```
FOR /L %V IN (1,1,10000) DO QUICK Create User=Test%V
```

The syntax for such DOS loop command is "For /L %V IN (start,step,end) DO [command]" where '%V' is the placeholder for the counter variable.

## 2.5 Variable Identifiers in Commands

Some QUICK for NDS options (see 'LoginScript', 'ForAll', 'ForSet') allow you to use paths, files, or complete command lines as parameters. Especially when these are used with wildcards, you may wish to somehow 'personalize' these parameters.

Example: you may want to save the user's login scripts to files, and you may consider using a command like

```
QUICK Loginscr User=* C:\test\LS.TXT
```

However, this command would simply save all scripts to the same file, thereby overwriting each other.

To save each script to a file with the users' names, you may use variables that will be replaced with the proper values at run-time and individually for each object. The command

```
QUICK Loginscr User=* C:\test\%name$.TXT
```

Would use the identifier '%name\$' as a placeholder and fill it with the individual object's name when the object is processed. If you would rather create file names using the users' surname instead of using their common names, you could use

*QUICK Loginscr User=\* C:\test\[Surname]\$.TXT*

Besides the name variable, there are a few other variable identifiers that help you personalize the command. Variable identifiers always are enclosed in '\$' characters.

\$name\$, or \$value\$	Object name (e.g., 'Admin'), or attribute value
\$type\$	Object type/class (e.g., 'User')
\$rdn\$	Relative distinguished name (e.g., 'CN=Admin')
\$dn\$	Distinguished name (e.g., 'CN=Lisa.OU=Org.O=Test')
\$context\$	The object's container (e.g., 'OU=Org.O=Test')
\$id\$	The object ID (e.g., 'EF006706')
\$date_ yyyy\$	Current year in 4-digit format (e.g., '2008')
\$date_ mm\$	Current month in 2-digit format (e.g., '01')
\$date_ dd\$	Current day in 2-digit format (e.g., '31')
\$date\$	Current date in 4-2-2 digit format (e.g., '2008-12-31')
\$time_ hh\$	Current hour in 2-digit format (e.g., '23')
\$time_ mm\$	Current minute in 2-digit format (e.g., '59')
\$time_ ss\$	Current second in 2-digit format (e.g., '59')
\$time_ ms\$	Current millisecond in 3-digit format (e.g., '999')
\$time\$	Current time in 2:2:2 digit format (e.g., '23:59:59')

In addition to these base variables you can retrieve any attribute (that can be accessed as string) from the selected object. This includes the standard attributes as well as new attributes from your own schema extensions. To access any attribute, simply enclose the name of the attribute in square brackets and Dollar signs ('\$[..]\$'), e.g., '\$[Surname]\$'.

## 2.6 Finding objects – the ‘Find’ option

Many QUICK commands allow to replace the “<otype>=<object>” format for selecting objects by a more sophisticated eDirectory search. This allows you to base your object selection on a powerful and individual search. A basic example of such a search could look like:

```
QUICK show "Find:((Base Class=User)&&(Surname=Sch*))"
```

This command would show you details on users who have a Surname that begins with ‘Sch’.

The find option is a replacement for the simple “<otype>=<object>” pattern matching search and can be combined with the regular command parameters, namely ‘/Sub’ and ‘/Cont=<name>’ to create subcontainer searches on a specified base container.

Since the “Find:” syntax usually contains blanks or reserved DOS symbols, you will typically include it in quotes.

The ‘Find’ syntax is extremely powerful and loosely based on syntaxes like SQL or LDAP searches. Here are some guidelines to create your own searches:

**Simple search filter:** A simple filter typically consists of an attribute name, an attribute operator, and an attribute value. Example: *"Surname=Miller"*

**Attribute names:** Attribute names can be any allowed object attribute as defined by the eDirectory schema (e.g., ‘Surname’, ‘Given Name’, ‘Internet EMail Address’). In addition to these, you may use special attribute names like ‘Base Class’ (identifying the base or primary class of the object) or ‘Name’ (referencing the naming attribute or relative distinguished name of the object).

Not all attribute types can be used in filters – the current implementation handles regular strings, integers, boolean and time attributes. If an unhandled attribute is used in a filter, the error ERR\_INVALID\_FILTER\_SYNTAX (-326) is returned.

**Attribute values and wildcards:** Attribute values may be whole strings (e.g., *"Surname=Miller"*), partial strings with wildcards (e.g., *"Surname=Mil\*"* or *"Surname=\*son"*) or plain wildcards (e.g., *"Surname=\*"*).

Boolean attributes can have the values ‘TRUE’ or ‘FALSE’. Date values must be entered in the format ‘yyyy-mm-dd’ or ‘yyyy/mm/dd’.

Note: eDirectory does not allow for wildcard searches on distinguished name attributes.

**Attribute operators:** The typical attribute operator will be ‘=’, but you may also use the operators ‘>=’ (greater or equal), ‘<=’ (smaller or equal), or ‘=\*’ (exists). Examples:

```
(Given Name=*)
(Surname=Smith*)
(manager=.admin.novell)
(Login Intruder Attempts>=3)
```

**Grouping:** Multiple simple search filters can be combined with filter operators, nested and grouped. For nesting and grouping simple filters, use the brackets ‘(’ and ‘)’. To combine two filters, simply place a filter operator between these two simple filters. Example:

```
((Surname=Doe)&(Given Name=J*))
```

**Filter operators:** You can combine by AND respectively OR-operations. Use “&&”, “&”, or “#AND#” to represent an **AND**-operation; use “|”, “|”, or “#OR#” to represent an **OR** operation. A **NOT**-operation is represented by “!” or “#NOT#”. Examples:

```
(Given Name=Jane) | (Given Name=Joe)
(Surname=Doe) #AND# (!(Given Name=John))
(Surname=Doe) && (Given Name=J*)
```

## 2.7 Error levels

If errors occur during program execution, QUICK will return an error level to DOS that depends on the error type. This allows more effective batch processing with QUICK. Also, the option 'Exists' explicitly uses the error level to indicate if the specified object exists, or not.

A simple demo for using the error levels could be a batch file containing these lines:

```
Quick exist Admin  
If errorlevel 1 echo "Admin, where are thou?"
```

## 2.8 Installation

When installing QUICK on a file server, follow the instructions on the installation disk. Failing to do so may result in improper operation of the software. At run-time you will need access to the executable and the DLLs. To allow starting QUICK independent of the workstation, it is recommended to store the DLLs in the same directory as the executable, or to copy the DLLs to the respective stations Windows system path.

## 2.9 Version History, Recent Changes

<b>QUICK for NDS</b>		
2006/09	V1.50x	<ul style="list-style-type: none"> <li>* added "Find:" option to allow powerful search and filter operations</li> <li>* added "/S" parameter to "exists" option to scan for objects</li> <li>* Option "ATTRIBUTE": added 'contains' operator</li> <li>* Option "ATTRIBUTE": added naming/mandatory attributes for all classes</li> <li>* Option "ATTRIBUTE": allow variables in new attribute values (e.g. "Full Name" = "\$[Given Name]\$ \$[Surname]\$")</li> <li>* Option "ATTRIBUTE": allow multiple attributes (e.g. "manager;surname") (get manager and surname)</li> <li>* Option "ATTRIBUTE": allow 'piped' attributes (e.g. "manager surname") (get manager's surname)</li> <li>* Option "ATTRIBUTE": allow special attributes creationTimestamp, modifyTimestamp, creatorsName, modifiersName</li> <li>* Added Date/Time placeholders</li> <li>* added option "HOMEDIR"</li> </ul>
2006	v1.41x	Additional improvements
2002-2005	v1.40x	Additional improvements Added more attribute decodes Changed license check: expiration information is stored in QUICK.INI
2001/07	v1.39	Fixed option to remove attribute values and group memberships. Prevent changing unhandled attributes Added option to allow hex change of Octet Strings attributes
2001/04	v1.38	Related objects' RDNs (e.g., group members) now depend on the defined context - see command line switch "/Cont=xxx". This allows for stronger control over the display format.
2000/12	v1.37	Added option "SERVER"
2000/11	v1.36	New general option /Alias=Ref/Deref New option "SECURITY"
2000/10	v1.35	Added find/replace for "LOGINSRIPT"
2000/09	v1.34	New options "CHECK", "PASSWORD", "RESTRICT" The "/Sub" parameter now available for most options
2000/06	v1.1 - v1.33	Each version added several new options
1999	v1.0	First release
<b>QUICK (Bindery)</b>		
1990-1996		QUICK – Bindery Tool to access NetWare Bindery Still available for NetWare 3.x under <a href="http://www.WolfgangSchreiber.de">http://www.WolfgangSchreiber.de</a>



## 3 QUICK for NDS: Available Options

Enter '*QUICK <option>*' (e.g., '*QUICK show*') to get a short syntax overview specific to the selected option.

See also:

→ p.16 General Options

### 3.1 Account: Check Object Account Information

#### Purpose

Retrieve or change account status information, including account balances, account restrictions like grace login or intruder status, and password restrictions

#### Syntax

```
QUICK Account [<otype>=]<object> [All|Balance|Enable|Disable]
```

#### Examples

```
QUICK Account someone.ou1.org  
QUICK Acc noone Balance  
QUICK Acc user=* All  
QUICK Acc user=test disable
```

#### Notes

While you can read several attributes that reflect the account restrictions and login status of an object, the only modification that currently is possible is disabling or enabling the account.

To change the other account related attributes like 'Login Expiration Time', 'Account Balance', use the 'Attribute' or 'Restrict' command options.

#### See also

Attribute: View or Change Object Attributes  
Restrict: Show or Change Object Restrictions  
Show: Show Object and Selected Properties  
→ p.16 General Options

### 3.2 Alias: View or Create Alias Objects

#### Purpose

View the Alias objects in a container, or create a new Alias to any object in a specified container.

If you specify two parameters after the 'Alias' option name, the first one is the existing object, and the second parameter specifies the new Alias name as a relative or absolute distinguished name.

If you just specify one parameter, QUICK will show the details for the selected alias.

In both cases, the first parameter may contain wildcards.



**Syntax**

*QUICK Alias <name\*> [<name>]*  
*"<name>" may contain variables enclosed in "\$" characters*  
*Avail variables: name, value, type, rdn, dn, context [<attr>]*

**Examples**

*QUICK Alias .someone.ou1.org*  
*QUICK Alias .someone.ou1.org .ab.cd.us*  
*QUICK Alias A\* .\$rdn\$.ou1.ou2.us*

**Notes**

You cannot combine wildcards and container names (e.g., ".A\*.cd.us") . To select multiple objects in a different container, use the "/cont" option, instead.

If you wish to create multiple Aliases at the same time, you may use wildcards; in this case you should use variables in the target name, since otherwise QUICK would try using the same alias name for each object.

Recursive operations are possible if you specify wildcards and use the "/sub" option. Again you will need to use variables in this case.

**See also**

Clone: Clone/Copy Existing Objects

Create: Create New Object

Show: Show Object and Selected Properties

➔ p.16 **General Options**

➔ p.**Error! Bookmark not defined. Variable Identifiers in Commands**

## 3.3 Attribute: View or Change Object Attributes

**Purpose**

You may view all or any specified attribute by simply entering the object name (may include wildcards) and the optional attribute name.

As usual, the attribute name must be enclosed in double quotes if it contains blanks or other special DOS characters (see p. 18 for details).

Be aware that some eDirectory attributes may have only one value (e.g., "Last Login Time"), while others may have multiple values (e.g., "Telephone Number"). This QUICK option allows you to maintain attributes according to their structure.

**Setting an attribute value:** Use the "=" operator to set and overwrite attribute values. If any attribute values existed, they will be removed, and the new value will be stored. Be careful not to inadvertently overwrite attribute values if you do not wish to.

**Adding an attribute value:** Use the "+" operator to add attribute values to the existing ones. If any attributes values existed, these will be remain unchanged, and the new value will be added to the existing attributes. Obviously, this will not work, if the eDirectory attribute is single-valued.

**Removing an attribute value:** Use the "-" operator to remove values from the existing set of attribute values. The value that you specify must exist in the current set, or you will receive an eDirectory error. If any other attributes values existed, these will be remain unchanged, and only the selected value will be removed.

**Checking an attribute value:** For flow control in scripts it may be necessary to check if an attribute contains a given value. You can use the 'contains' or '!contains' (contains not) operators to perform such a check. When using these operators an errorlevel of '0' will be returned if the condition is true; a higher error level will be returned if the condition is false. The comparison is not case-sensitive. For multi-valued attributes, 'contains' returns 'true', if any of the values contains the given string.

Since the simple QUICK call to retrieve an attribute will return an errorlevel if there is no value, you can use this feature to check for the pure existence of an attribute value.

**Example**

```
QUICK Attr .oliverT.org Surname contains twist
If errorlevel 1 echo "not that oliver"
QUICK Attr .oliverT.org pager
If errorlevel 1 echo "oliver has no pager"
```

**Using variables in attribute values:** When setting/adding/removing attribute values you may use variables and placeholders in the replacement string. This allows you to set context sensitive attribute values parameters (e.g., `QUICK Attr User=Test* "Full Name" = "$[Given Name]$ ${Surname}$"`).

**Synchronizing attribute values** If the user attribute "Group Membership" or the group attribute "Member" is changed with the '+' or '-' operator), QUICK will automatically synchronize the related backlink attribute and update the security equivalencies.

**View default attributes:** To view some default attributes of an object, you may simply call this option with the target object(s) and no further parameters (e.g., `QUICK Attr User=Test*`). QUICK will use a set of standard attributes to display the objects.

**View special attributes:** Some 'pseudo' attributes may also be viewed (but not modified). These pseudo attributes include "creatorsName", "createTimeStamp", "modifiersName", "modifyTimestamp", "CONTEXT", "CLASS". You can use these pseudo attributes in the "QUICK Attribute" command like regular attributes (e.g., `QUICK Attr User=Test* creatorsName`).

**View multiple attributes:** To view a custom set of object attributes of, you may simply specify the target attributes, separated by semicolon. If the attribute names contain blanks, you need to enclose them in quotes (e.g., `QUICK Attr User=Test* "Given Name;Surname;Manager"`).

**'Piping' attributes:** You may use "QUICK Attribute" not only to get the selected object's attributes, but also the attributes of interlinked objects. To get the Surname of an object's manager, you may use a command syntax like `QUICK Attr User=Test* manager/Surname`). Such 'piping' of attributes will only work for single-valued attributes that contain distinguished names. Since the pipe symbol '|' is a reserved character in DOS, you will always need to enclose such piped attributes in quotes. You may even create longer pipes like like `QUICK Attr User=Test* manager/assistant/Telephone Number` – this command will return the telephone number of the assistant of the manager of the selected object. This feature may be combined with the option to view multiple attributes: to get the first and last name of an object's manager, use a command line like `QUICK Attr User=Test* "manager/Given Name;manager/surname"` – the semicolon has precedence over the pipe symbol.

**Syntax**

```
QUICK Attribute <name*> [<attr> CLEAR/[+|-/= <value>]]
```

**Examples**

```

QUICK Attribute .someone.oul.org
QUICK Attribute .ab.cd.us "Department"
QUICK Attribute A* "Department" + something
QUICK Att User=* "Full Name" = "King $[Surname]$"
QUICK Att User=* "Given Name;Surname"
QUICK Att User=* "manager|Surname"

```

"<value>" may contain variables enclosed in "\$" characters  
 Avail variables: name, value, type, rdn, dn, context, [<attr>]

## Notes

**Attribute Names:** To get more information about any attributes, e.g., what syntax it has and if it is single- or multi-valued, use the QUICK option "eDirectory: Retrieve eDirectory Info".

Be careful not to inadvertently overwrite or remove attribute values.

Clear: the option "Clear" allows you to clear all values of an attribute.

**Time/Date:** The time/date format may depend on your local workstation settings, e.g. "yyyy-mm-dd hh:mm:ss" or "mm/dd/yy hh:mm:ss". While eDirectory internally saves time stamps as GMT, QUICK will do a conversion to local time.

**Counters:** eDirectory classifies some attributes as 'Counters', e.g., the attribute "Login Grace Remaining". For these counters, this QUICK command will work slightly different: '=' will set the counter to the specified value, '+' will increment, and '-' will decrement the counter by the specified value.

**Restrictions:** While entering attribute values with this QUICK option works fine for the basic attributes like strings, numbers, counters, fax/telephone numbers - more complex attributes (network addresses, streams, lists, etc) are not supported here. Paths are only supported if entered in their DOS short format (no long names). Boolean variables (e.g., "Login Disabled") can be entered as 'TRUE'/'T'/'YES'/'Y' while any other input counts as 'FALSE'. Octet Strings can be entered as strings, or - preceded by '0x' - as hex data (e.g., '0x0102ABFF'), but be aware of the maximum DOS command line length.

The ACL attribute is a structured attribute that consists of 3 elements: trustee name, subject, and rights. If you plan to modify the ACL attribute, you need to pass these 3 elements as one string in the format '<trustee>#<subject>#<rights>'. Replace <trustee> by the name of the object that you want to grant rights to (e.g., '.someone.oul.org', '[Public]', or '[Self]'). Replace <subject> by the scope of the trustee (e.g., '[Entry Rights]', '[All Attributes Rights]', or the name of an attribute). Replace <rights> by the requested trustee rights; you may enter a numeric value (equivalent to the ones in LDIF files), or the initial letters for the requested rights. For granting trustee rights to attributes, these rights are available: Compare, Read, Writes, Self, Admin, Inherit - for granting trustee rights to '[Entry Rights]', these rights are available: Browse, Add, Delete, Rename, Supervisor, Inherit). As usual, you need to enclose the composed string in quotes if it contains blanks.

Sounds difficult enough, so here are some examples:

```

QUICK Attr .ab.us ACL + ".someone.org#[Entry Rights]#BADRST"
QUICK Attr .ab.us ACL + [Self]#Language#RWAI

```

**Warning:** by changing the ACL, you may inadvertently restrict access rights and prevent access to objects, thereby excluding yourself and others from accessing the objects. Before working in a production environment, be sure to thoroughly test the syntax in a lab tree.

On [http://developer.novell.com/ndk/doc\\_ndslib.htm](http://developer.novell.com/ndk/doc_ndslib.htm) you can get more information on the

eDirectory schema, objects and attributes. You may also use the "QUICK eDirectory" command to get information on available classes and attributes.

**See also**

Groups: Show or Change Group Memberships

Member: Show or Change Group Members

eDirectory: Retrieve eDirectory Info

Restrict: Show or Change Object Restrictions

Rights: Get Object's Rights

→ p.16 **General Options**

→ p.**Error! Bookmark not defined. Variable Identifiers in Commands**

## 3.4 Check: Inspect Object Integrity

**Purpose**

Perform an eDirectory integrity check on one or more objects. This option queries eDirectory to retrieve any errors that eDirectory has stored for this object.

If eDirectory knows about errors that occurred with the selected objects, the error messages will be returned.

You can use the "/sub" option to check whole branches of your eDirectory tree.

**Syntax**

*QUICK Check [<type>=<name>*

**Examples**

*QUICK Check User=\* /sub*

**Notes**

When checking multiple objects using wildcards it may be helpful to redirect output with DOS pipes, or to use the /Quiet option (see 'General Options' on p.12) to restrict output to errors.

**See also**

List: List Objects

Show: Show Object and Selected Properties

→ p.16 General Options

## 3.5 Clone: Clone/Copy Existing Objects

**Purpose**

Create a copy of the existing object.

**Syntax**

```
QUICK Clone <obj> <obj>/<container> [/sub]
```

**Examples**

```
QUICK Clone .someone.ou1.org noone  
QUICK Clone noone ou1.ou2.us  
QUICK Clo .test.ou1.org ou2.org /sub
```

**Notes**

Syntax and functionality of this function are identical with the COPY option. See there for further details.

**See also**

Exists: Check Object Existence  
Copy: Clone/Copy Existing Object  
Move: Move Objects to a Different Container  
→ p.16 General Options

## 3.6 Context: Get or Set the Default Context

**Purpose**

Unless you explicitly specify a tree or eDirectory context in your QUICK command, the command will determine the default tree and context which is usually defined in your Novell Client settings. This approach is comparable to the concept of the current drive and path in the file system for DOS commands. QUICK will identify these defaults and use them as base for executing the requested command.

To work with different trees and/or contexts, you may specify these in the command, e.g., in the form of command line parameters '/Tree=xxx' and '/Context=xxx'.

If you frequently work on trees or contexts that are different from your defaults, you may change these with the "QUICK Context" command.

**Syntax**

```
QUICK Context Set/Get
```

**Examples**

```
QUICK Context Get  
QUICK Context Set /cont=.myOu.myOrg /tree=MyTree
```

**Notes**

The changed defaults will persist until the station is rebooted.

**See also**

→ p.16 General Options

## 3.7 Copy: Clone/Copy Existing Objects

### Purpose

Create a copy of the existing object. The original object will be read, and all accessible information copied to the target object.

You may specify a target container (then the new object will be generated with the same name but in a different container). You may also specify a new object name in the current or in a different container (then the new object will be generated with the specified name in the same or in a different container).

If you specify a container as a source object, you may copy/clone complete containers including their sub-containers.

### Syntax

```
QUICK Copy <obj> <obj>/<container> [/sub]
```

### Examples

```
QUICK Copy .someone.ou1.org noone  
QUICK Copy noone ou1.ou2.us  
QUICK Co .test.ou1.org ou2.org /sub
```

### Notes

You may copy leaf objects (like users, groups, etc) or container objects (like Organizational Units). The option '/sub' obviously can only be used if the source object is a container.

To include the child objects in the cloning process (clone containers and it's leaf nodes and sub-containers) you may specify the '/sub' option. If a whole container with its child objects is cloned, the function will clone the individual objects one by one. Cloning containers with larger numbers of child objects may take a while, but QUICK for NDS will display the status.

Wildcards: Wildcards are not supported. To copy multiple objects use the '/sub' option.

If the specified target object exists, it must be a container – the source object will be copied to that container.

If the specified target object does not exist, it will be created with the same object type as the source object.

Changing the name while cloning: If you specify an existing target container, the cloned object will be created with the same name in the new container.

The syntax is kept similar to the COPY command that you know from DOS: If you specify a new name that does not yet exist, the object will be created with that new name.

As a sample, take the command "*QUICK Copy obj1 obj2*":

If obj2 is an existing container, QUICK will create a copy of obj1 in container obj2 without changing the name. If obj2 does not exist, QUICK will create a copy of obj1 with the new name obj2.

Hint: You should only clone objects if you have administrative rights to the source object and the target container.

Internally this function calls a backup API to read the object information, and a the equivalent restore function to save it as the target object.

Caution: Before using the cloned objects in a production environment, you should carefully test them. The 'Clone/Copy' option will create exact duplicates of the source objects. The behavior of the 'GUID' attribute may depend on the eDirectory version that you use. Also, for DirXml/IDM users it is important to know, that the association attribute will be identical for the source and the clone. This effect may be desired, but you may

also remove the clone's association with the respective 'QUICK Attr' command.

**See also**

Clone: Clone/Copy Existing Object

Copy: Clone/Copy Existing Objects

Delete: Delete Objects

Exists: Check Object Existence

Move: Move Objects to a Different Container

→ p.16 **General Options**

→ p.**Error! Bookmark not defined. Variable Identifiers in Commands**

## 3.8 Create: Create New Object

**Purpose**

Create new users, groups, or containers. The current version of QUICK for NDS will only allow creation of these standard objects.

If you need to create additional object types, please email the author.

**Syntax**

*QUICK Create <otype>=<name>*

**Examples**

*QUICK Create User=.someone.ou1.org*

*QUICK Create Group=.someone.ou1.org*

*QUICK Create "Organizational Unit"=something*

**Notes**

If the specified object exists, already, an error message is returned. The current version of QUICK for NDS does not allow using templates for user creation.

New users will have their user name as surname, and an empty string as password.

**See also**

Attribute: View or Change Object Attributes

Clone: Clone/Copy Existing Object

Move: Move Objects to a Different Container

Exists: Check Object Existence

→ p.16 General Options

## 3.9 Delete: Delete Objects

### Purpose

Delete any existing object.

### Syntax

*QUICK Delete [<otype>=]<name\*>*

### Examples

*QUICK Delete .someone.oul.org*  
*QUICK Delete User=A\**

### Notes

This function will delete single or multiple objects of any kind. Be aware that QUICK for eDirectory does not ask for confirmation before removing the object. It is strongly recommended, especially when using wildcards, to assure that you have a valid eDirectory backup, so that inadvertently deleted objects can be recovered.

### See also

Copy: Clone/Copy Existing Objects  
Clone: Clone/Copy Existing Object  
Exists: Check Object Existence  
Move: Move Objects to a Different Container  
→ p.16 General Options

## 3.10 Dump: Show All Available Object Attributes

### Purpose

The Dump option allows you to view all accessible information on one or multiple objects. The function will retrieve all attributes for the given object type, and try to read and display all attribute values. This includes values from your own schema extensions. Not all attributes may be displayed in readable format – QUICK for NDS tries to display them in readable output format, but unrecognized attribute values may be displayed as hexadecimal strings.

The primary purpose of this option is to analyze any given object and its attributes.

### Syntax

*QUICK Dump [<otype>=]<name\*>*

### Examples

*QUICK Dump User=A\**  
*QUICK Dump .someone.engineering.novell*

### Notes

DUMP is included primarily for testing and programming purposes. All available information will be displayed.

If an optional attribute does not exist or does not have a value for the object, QUICK for eDirectory will display the value as [n/a]. The same will be displayed if the attribute is present, but cannot be read due to lack of rights.

### See also

Attribute: View or Change Object Attributes



List: List Objects  
Show: Show Object and Selected Properties  
→ p.16 General Options

## 3.11 eDirectory: Retrieve eDirectory Info

### Purpose

This option allows you to retrieve information about your tree and schema. The parameter 'Classes' lists all classes (i.e., object types) that exist on your tree. The parameter 'Attributes' lists all attributes (i.e., object types) that exist for the specified class (object type). To get a complete overview of all your classes and all of their attributes, you can use the option 'Schema'.

All parameters respect and handle the standard eDirectory schema as well as your own schema extensions.

### Syntax

```
QUICK eDirectory Classes | Attributes <otype> | Schema
```

### Examples

```
QUICK eDir Classes  
QUICK eDir Attrib "Organizational Unit"
```

### Notes

You may use this option to document your tree's schema before and after installing 3<sup>d</sup> party applications that extend the schema. From the different output, you can identify the schema extensions made by the installation.

### See also

Attribute: View or Change Object Attributes  
Dump: Show All Available Object Attributes  
→ p.16 General Options

## 3.12 Equivalencies: List Security Equivalencies

### Purpose

View or change the list of security equivalencies for any object. Security equivalencies are used by eDirectory to grant one objects the same rights as another object. They are stored in the eDirectory attribute "Security Equals". To grant UserX the same rights as Admin, you may add Admin to the list of security equivalencies for UserX. Using the '+' and '-' options, you may change the equivalencies of any object.

### Syntax

```
QUICK Equival [<otype>=]<name*> [+/- [<otype>=]<name>]
```

### Examples

```
QUICK Equ User=A*
QUICK Equ .everyone.us + .someone.ou1.org
```

**Notes**

When making User1 equivalent to User2, eDirectory will save the name of User2 in the attribute "Security Equals" of User1. At the same time eDirectory will add the name of User1 in the attribute "Equivalent To Me" of User2.

The attribute "Equivalent To Me" contains a list of entries that are security equivalent to the entry containing the attribute.

The attribute "Equivalent To Me" specifies group membership and security equivalencies of an object.

If you run this option without any parameters, QUICK for NDS will display both perspectives: it will show the contents of the "Equivalent To Me" and the "Security Equals" attributes.

To check who has been granted Admin equivalencies, run

```
QUICK Equ Admin
```

To make UserX Admin equivalent, run

```
QUICK Equ UserX + Admin
```

**See also**

Groups: Show or Change Group Memberships

Member: Show or Change Group Members

Rights: Get Object's Rights

→ p.16 General Options

## 3.13 ErrMsg: Convert Error Number Into Message

**Purpose**

If an error occurs during QUICK for NDS operation, QUICK for NDS will try returning a more meaningful error text instead of a cryptic error code. Many eDirectory tools and server applications, however, return only error codes if a problem occurs.

If you have another EDirectory application that returns such cryptic error codes, you can use this option to convert the error code into a more meaningful error message, thus helping you identify the problem.

The error code can be entered as decimal number, or as hex number preceded by '0x'.

**Syntax**

```
QUICK ErrMsg <error number>
```

**Examples**

```
QUICK Err 0x899C
```

```
QUICK Err -601
```

**Notes**

The error codes are converted by NWERR32.DLL that comes with QUICK for NDS

**See also**

→ p.16 General Options

## 3.14 Exists: Check Object Existence

### Purpose

Check if a specified object exists. This option can be used from the command line, but is especially useful in batch jobs if you need to check if an object exists before doing any other operations. If the object exists, QUICK for NDS will return an errorlevel of 0, but since the option returns an errorlevel if the object does not exist, you can act accordingly in subsequent commands.

Typically this will be done by a sequence of commands like

```
QUICK Ex user=.newuser.test.us
If errorlevel 1 goto label4
```

### Syntax

```
QUICK Exists [<type>=<name>
```

### Examples

```
QUICK Exists User=.someone.ou1.org
```

### Notes

Used in batch files with the subsequent DOS command "IF ERRORLEVEL 1 ..."

### See also

List: List Objects  
Show: Show Object and Selected Properties  
→ p.16 General Options

## 3.15 ForAll: Execute Command for Matching Objects

### Purpose

This option is extremely powerful if you need to execute any command for each object, or for a subset of objects.

Filters: The object filter ([<type>=<name\*>) specifies the set of objects that you want to select.

Command line: The "cmd" can be any DOS or Windows executable or batch file that will be executed once for each object that matches the filter condition.

The command

```
QUICK User=Test* "Echo $name$"
```

will run the command "Echo \$name\$" for each user with a name matching the filter condition

Variables: You will typically need to 'personalize' the command line. This can be done with a set of variable identifiers, which will be replaced with user specific information before the command line is called. In the sample above, the variable "\$name\$" will be replaced by the user's name (e.g., 'Test1', 'Test2', etc.).

See details on variable identifiers in the chapter '**Variable Identifiers in Commands**' on page **Error! Bookmark not defined.**

Modifiers: Like with most other QUICK for NDS options you may add optional parameters that specify the starting tree and container, or additional display options. See details in the section on '**General Options**' on page 16.

The most useful option here will be '/sub' which allows you to recurse through sub-containers.

**Syntax**

```
QUICK ForAll [<type>=<name*> "cmd"
```

Avail variables: \$name\$, \$value\$, \$type\$, \$rdn\$, \$dn\$ \$context\$ \$[<attr>]\$

**Examples**

```
QUICK ForAll A* "echo $name$ $[mailstop]$" /Cont=.ou1.ou2.us /su
```

```
QUICK ForAll User=A* "md F:\users\$name$" /qu
```

**Notes**

The command line passed to QUICK for NDS needs to be enclosed in double quotes since it usually includes blanks.

When using the variable identifier to retrieve any attribute, be aware that some attributes may be multi-valued (like group memberships) or may contain complex information (like login scripts). These complex attributes may create a command line that cannot be interpreted by DOS/Win.

Before running the QUICK with this option, you might test the command by simply preceding the command with an 'REM ' or "ECHO " and scrutinize the resulting output before actually running the command.

Using internal DOS commands:

As indicated above, the command can be a DOS or Windows executable or batch file.

To use internal DOS commands like 'md', 'rd', or 'del', simply use a text editor to create a batch file (e.g., "DOS.bat") with this contents:

```
%1 %2 %3 %4 %5 %6 %7 %8
```

```
pause
```

Then use this batch file to call internal DOS commands, e.g.:

```
QUICK forall user=r* "DOS.bat md $name$"
```

**See also**

ForSet: Execute Command for Each Member of Attribute Set

→ p.16 **General Options**

→ p.**Error! Bookmark not defined. Variable Identifiers in Commands**

## 3.16 ForSet: Execute Command for Each Member of Attribute Set

**Purpose**

This option is very powerful if you need to execute a command for each member of a given set.

'Sets' are multi-valued attributes like 'Group Members', 'ACL', or 'Security Equals', and – similar as with the 'ForAll ' option – any command can be executed for each object in this set.

You might for example call a command for each object that is member of a specified group.

Filters: The object filter ([<type>=<name\*>) specifies the set of objects that you want to select.

Attribute: Specify the set (the multi-valued attribute) that you want to execute the command for. Typical sets include "Security Equals", "Members", "ACL", "CN", and "Group Member". Each class or object type will own a specific range of attributes, as defined in the eDirectory schema. To get a list of available attributes for a given object type call

```
"QUICK eDirectory Attrib <otype>
```

e.g.,

```
QUICK eDirectory Attrib "User"
```

See there for further details on the schema options of QUICK for NDS.

Command line: The "cmd" can be any DOS or Windows executable or batch file that will be executed once for each object that matches the filter condition.

The command

```
QUICK User=Test* "Echo $name$"
```

will run the command "Echo \$name\$".

Variables: You will typically need to 'personalize' the command line. This can be done with a set of variable identifiers, which will be replaced with user specific information before the command line is called. In the sample above, the variable "\$name\$" will be replaced by the user's name (e.g., 'Test1', 'Test2', etc.).

See details on variable identifiers in the chapter on '**Variable Identifiers in Commands**' on page **Error! Bookmark not defined.**

Modifiers: Like with most other QUICK for NDS options you may add optional parameters that specify the starting tree and container, or additional display options..

The most useful general option here will be '/sub' which allows you to recurse through sub-containers.

### Syntax

```
QUICK ForSet [<type>=]<name*> <attr> "cmd" [/quiet]
```

Available variables: \$name\$, \$value\$, \$type\$, \$rdn\$, \$dn\$

### Examples

```
QUICK ForSet A* "Security Equals" "echo $rdn$"
```

```
QUICK ForSet .ou1.ou2.us ACL "type $value$" /qu
```

### Notes

The command line passed to QUICK for NDS needs to be enclosed in double quotes since it usually includes blanks.

When using the variable identifier to retrieve any attribute, be aware that some attributes may be multi-valued (like group memberships) or may contain complex information (like login scripts). These complex attributes may create a command line that cannot be interpreted by DOS/Win.

Before running the QUICK with this option, you might test the command by simply preceding the command with an 'REM ' and scrutinize the resulting output before actually running the command.

Using internal DOS commands: See notes with "ForAll"

### See also

ForAll: Execute Command for Matching Object

→ p.16 **General Options**

→ p.**Error! Bookmark not defined. Variable Identifiers in Commands**

## 3.17 Groups: Show or Change Group Memberships

### Purpose

This option allows you to view and change group memberships of any object.

To just view the memberships, call the option with just the name of the selected object.

To add or remove groups from the membership list, specify the group object's name with '+' or '-' to add or remove the group from the list.

The option will update both, the 'Group Membership' and 'Security Equals' attributes of the object, as well as the 'Member' and "Equivalent To Me" attributes of the group.

### Syntax

```
QUICK Groups [<otype>=]<name*> [+|- <group>]
```

### Examples

```
QUICK Groups .someone.ou1.org
```

```
QUICK Groups User=A* + .everyone.us
```

### Notes

Both, the object and the group must exist.

The user name may include wildcards. If you want to add/remove memberships, the group name must be unique.

eDirectory does not perform any reasonability check, but simply update the membership list with whatever you pass as a parameter. It is your responsibility to add only group memberships that make sense.

### See also

Attribute: View or Change Object Attributes

Groups: Show or Change Group Memberships

➔ p.16 General Options

## 3.18 HomeDir: Handle Home Directories

### Purpose

Show, create, or delete user home directories.

To appropriately take care of user home directories, you need to consider three aspects: a) the existence of the home **directory** in the file system on a server, b) the appropriate trustee **rights** of the user, and c) the user's "Home Directory" **attribute**.

When using the QUICK options "homedir create/delete", QUICK will update all three of these aspects, unless you explicitly exclude one or more aspects.

In the case of home directory creation, the command

```
QUICK Homedir User=Test* Create [MYSRV\MYVOL:HOME]
```

will create a directory with the CN of the current user under the specified NetWare path, then assign all rights except Administrator rights to the user, and add the path to the user's "Home Directory" attribute. To prevent one of these three steps, you may add one or more of the command line options "/noDirectory", "/noRights", or "/noAttribute".

If you add the "/noDirectory" option, QUICK will not create a new directory.

If you add the "/noAttribute" option, QUICK will not update the "Home Directory" attribute.

If you add the “/noRights” option, QUICK will not grant trustee rights to the directory. To specify a different set of rights, use the “/rights=” parameter; here you may specify a combination of the rights RWCEAFMS, e.g. “/Rights=RCWEFMS”

If the user already has a valid “Home Directory” attribute, you may omit the <PATH> parameter, since QUICK will then use the contents of that attribute. This allows you to fix situations, where the eDirectory information does not match the file system paths and trustee rights, or where you want to modify the home directory trustee assignments. A command like

```
QUICK Home User=* Create /rights=RWCEAMF /noAttr /noDir
```

would set the new trustee assignments, but not create directories or change the “Home Directory” attribute

In the case of home directory deletion, the command

```
QUICK Homedir User=Test* Delete [MYSRV\MYVOL:HOME]
```

will delete the directory with the CN of the current user under the specified NetWare path, then remove the user’s trustee rights, and finally remove the user’s “Home Directory” attribute. Again, you may prevent some of these three steps by adding a command line options “/noDirectory”, “/noRights”, or “/noAttribute”.

If the user has a valid “Home Directory” attribute, you may omit the <PATH> parameter, since QUICK will then use the contents of that attribute.

To simply display the users’ “Home Directory” attribute contents, use a command format like

```
QUICK Homedir User=*
```

### Syntax

```
QUICK Homedir [<otype>=<name*>] [create/delete] [<PATH>]
[/noRights] [/noDirectory] [/noAttribute]
```

### Examples

```
QUICK Homedir User=*
QUICK Homedir User=* create Srv2\Vol1:Home
QUICK Homedir User=* create Srv2\Vol1:Home /noRights
QUICK Homedir User=* create /noAttr
QUICK Homedir User=* delete
```

### See also

- List: List Objects
- Show: Show Object and Selected Properties
- p.16 General Options

## 3.19 List: List Objects

### Purpose

List the names of objects in a given container. You may restrict output by applying a name or object type restriction.

### Syntax

```
QUICK List <otype>[=<name*>]
```

### Examples

```
QUICK List User
```

```
QUICK List User=A*
QUICK List *
QUICK List User /Cont=.ou1.ou2.us
```

**Notes**

Use the format `QUICK List *=*` or `QUICK List *` to list all objects in a container.

To list only group objects, you could use the format `'QUICK List Group'`.

To list all users starting with the letter 'A', you can use `'QUICK List User=A*'`

To list any object starting with the letter 'A', you can use `'QUICK List *=A*'`

If the type name includes blanks, do not forget to enclose it in double quotes, e.g. `'QUICK List "organizational unit"=A*'`

**See also**

Attribute: View or Change Object Attributes  
Show: Show Object and Selected Properties  
→ p.16 General Options

## 3.20 LoginScript: Display Login Scripts

**Purpose**

This options allows you to view, save, or upload login scripts for any object. The objects typically are users, templates, containers, or profiles.

If you just pass the object name, QUICK for NDS will display the script, i.e., the default parameter is 'Show'. Optionally you may save the script to a plain text file specified in the command line, or upload the script from a file to the object's script.

You may for example use this option to copy scripts from one object to another, or to edit the script with standard test editors.

If you use the option with an object filter using wildcards and save or upload scripts, you may wish to 'personalize' the command and use different file names for each object.

You may simply do so by using variable identifiers in the path/file name: you could for example use the identifier '\$name\$' in the path or file name, and this placeholder would be replaced by each object's common name at run-time.

See details on variable identifiers in the chapter on '**Variable Identifiers in Commands**' at page **Error! Bookmark not defined..**

You may programmatically search or replace text in login scripts. The search is not case sensitive, and during the replace operation all matching text occurrences will be replaced with the new string without confirmation. To be on the save side with bulk change operations, you might use an eDirectory backup or the "save" option of this QUICK operation to have the option to undo your operation where necessary.

The "save" and "load" options take the file name as parameter, the "find" option takes the search string as only parameter, and the "replace" option requires two additional parameters: the search text and the replace text.

**Syntax**

```
QUICK LoginScript [<otype>=]<name*> [show/find/replace/save/load]
```

**Examples**

```
QUICK LoginScript .someone.ou1.org
QUICK Log .someone.ou1.org load C:\x.txt
```



```
QUICK Log User=A* save C:\$name$.txt
QUICK Log A* find "something"
QUICK Log User=A* replace "MAP X:" "MAP Y:"
```

**Notes**

Login scripts may contain many lines of script commands. Be sure to have sufficient disk space available if you save the scripts. Also the display of scripts may scroll over the screen faster than you can read.

**See also**

Attribute: View or Change Object Attributes

Show: Show Object and Selected Properties

→ p.16 **General Options**

→ p.**Error! Bookmark not defined. Variable Identifiers in Commands**

## 3.21 Member: Show or Change Group Members

**Purpose**

This option allows you to view and change the members of groups.

To just view the group members, call the option with just the name of the selected group.

To add or remove groups from the membership list, specify the member's object name with '+' or '-' to add or remove the member from the list.

The option will update both, the 'Group Membership' and 'Security Equals' attributes of the object, as well as the 'Member' and "Equivalent To Me" attributes of the group.

**Syntax**

```
QUICK Member [<otype>=]<name*> [+/- [<otype>=]<name>]
```

**Examples**

```
QUICK Member .everyone.us
```

```
QUICK Member .everyone.us + .someone.oul.org
```

**Notes**

Both, the object and the group must exist if you wish to add/remove members.

The group name may include wildcards. If you want to add/remove members, the member name must be unique and must not contain wildcards.

eDirectory does not perform any reasonability check, but simply updates the membership list with whatever you pass as a parameter. It is your responsibility to add only group memberships that make sense.

**See also**

Attribute: View or Change Object Attributes

Groups: Show or Change Group Memberships

→ p.16 **General Options**

## 3.22 Move: Move Objects to a Different Container

### Purpose

This object can be used to move individual or multiple objects to a different container

### Syntax

```
QUICK Move [<otype>=]<name> <container> [/sub]
```

### Examples

```
QUICK Move .someone.ou1.org .ou1.ou2.us  
QUICK Move .ou1.ou2.us .ab.cd.us /su
```

### Notes

You should not move objects that are currently logged in.

You may move leaf objects (like users, groups, etc) or container objects (like Organizational Units). The option '/sub' obviously can only be used if the source object is a container. Containers can be moved, either if they do not have any child objects, or if you move them with all child objects ('/sub' option)

To include the child objects in the move process (move containers and it's leaf nodes and sub-containers) you may specify the '/sub' option. If a whole container with its child objects is moved, the function will move the individual objects one by one. Moving larger containers may take a while, but QUICK for NDS will display the current status.

Wildcards: Wildcards are not supported. To move multiple objects, use the '/sub' option.

The target object must be a container.

You should only move objects if you have administrative rights to the source object and the target container.

Before using the moved objects in a production environment, you should carefully test them.

Limitation: After a container is moved with subdirectories, sometimes the move to the new target works correctly but the now obsolete original container structure may still remain as a copy without any leaf objects.

This is due to eDirectory limitations: QUICK first copies the container structure to the new location, then moves all leaf objects, before deleting the obsolete original structure. If synchronization is slow, eDirectory prevents deletion of the obsolete structure until synchronization is complete. If QUICK cannot remove that structure within 10 seconds after the move, it will return. You may manually delete the obsolete structure.

### See also

Clone: Clone/Copy Existing Object  
Copy: Clone/Copy Existing Objects  
Exists: Check Object Existence  
Delete: Delete Objects  
Rename: Rename Existing Object  
➔ p.16 General Options

## 3.23 Options: List Available Command Line Options

### Purpose

List the available QUICK for NDS command line options with short descriptions.

### Syntax

QUICK Options

### Examples

*QUICK Opt*

### Notes

This option does not have any additional parameters.

### See also

Syntax: Show All QUICK Syntaxes

## 3.24 Password: Change Object Password

### Purpose

Change an object's password.

### Syntax

*QUICK Password [<otype>=]<name\*> <pwd> [oldpwd]*  
*QUICK Password [<otype>=]<name\*> ? <pwd>*

### Examples

*QUICK Password User=A\* something*  
*QUICK Password User=A\* = secret*  
*QUICK Password .someone.ou1.org something anything*

### Notes

If you have administrative rights on the object, you do not need to specify the old password of the object. In this case, QUICK for NDS will generate a new object key pair in eDirectory, which may take slightly longer than the process if you do specify the old password.

If you have administrative rights on the object, and you specify the old password, but an incorrect one, first the password change is tried with the assumed old password, then – if that step fails - the command is treated as if you hadn't specified the old password and a new object key pair is generated.

Why then specify the old password at all? Just because the process is slightly faster, but the effect may only be noticeable if you change passwords for a larger number of objects.

You may also use this option to verify a given password by using the syntax

*QUICK Password [<otype>=]<name\*> ? <pwd>*

### See also

Copy: Clone/Copy Existing Objects

→ p.16 General Options

## 3.25 Rename: Rename Existing Object

### Purpose

Rename an existing object to give it another name. The relative distinguished name will be changed to the new name.

### Syntax

```
QUICK Rename [<otype>=]<name*> <name>
```

### Examples

```
QUICK Rename .OHardy.ou1.org SLaurel  
QUICK Rename User=A* New_&name&
```

### Notes

The new object name must be unique and must not yet exist in the container. To move and rename an object at the same time, you should use the MOVE option.

You may use wildcards in the Rename command, but you should use variable placeholders in the new name, or only the first object will be renamed.

### See also

- Clone: Clone/Copy Existing Object
- Copy: Clone/Copy Existing Objects
- Delete: Delete Objects
- Exists: Check Object Existence
- Move: Move Objects to a Different Container

➔ p.16 General Options

## 3.26 Restrict: Show or Change Object Restrictions

### Purpose

Display or change existing object restrictions. Typically the object will be of type User.

You must specify the object name that may include the object type and a wildcard. The 3<sup>rd</sup> parameter specifies the restriction that you want to view or change, and the optional 4<sup>th</sup> parameter specifies the new value of the restriction. Depending on the restriction, the value may be a number, a string, or a Boolean value (True/False, T/F, Yes/No, Y/N). These restrictions may be used (you may use abbreviations as long as you keep them unique):

Restriction	Effect	Parameter
<i>balance</i>	Account balance	[Number]
<i>disable</i>	Account disabled	[Boolean]
<i>graceLogin</i>	Number of grace Logins	[Number]
<i>lock</i>	Account locked	[Boolean]
<i>maxConn</i>	Maximum simultaneous connections	[Number]
<i>pwdChg</i>	Allow user to change password	[Boolean]
<i>pwdExp</i>	Password expiration interval	[Number]
<i>pwdLen</i>	Minimum password length	[Number]
<i>pwdReq</i>	Password required	[Boolean]
<i>pwdUnique</i>	Unique password required	[Boolean]
<i>volSpace</i>	Volume space restriction	<srv>/<vol:>[=<kB>]

While the other restrictions do not require the 4<sup>th</sup> parameter when viewing information, you need to specify the server and volume when viewing volume restrictions with the option *volSpace* (e.g., "*QUICK Restrict User=\* volsp FS3/Sys:*" to view or "*QUICK Restrict User=\* volsp FS3/Sys:=1024*" to change volume space restrictions.

### Syntax

```
QUICK Restrict [<otype>=]<name*> [<restriction> [value]]
```

Available restrictions:

```
balance, disable, graceLogin, lock, maxConn  
pwdChg, pwdExp, pwdLen, pwdReq, pwdUnique, volSpace
```

### Examples

```
QUICK Restrict User=.someone.oul.org Balance
```

```
QUICK Restrict User=A* pwdUniq true /su
```

### Notes

Many features of this option can also be accessed with the options "Account: Check Object Account Information" (p.25) or "Attribute: View or Change Object Attributes" (p.26).

### See also

Account: Check Object Account Information  
Attribute: View or Change Object Attributes  
Show: Show Object and Selected Properties  
→ p.16 General Options

## 3.27 Rights: Get Object's Rights

### Purpose

View the eDirectory rights of the specified object.

If you use the option in its base format without specifying a second object, the command will display the ACL list of the selected object. Here you will see all rights that the object has been granted to other objects or attributes in the tree.

You may specify a second object and optional object attributes, and in this case the command will display the effective rights to that second object or attribute.

### Syntax

```
QUICK Rights [<type>=]<name*> [[<type>=]<name> [<attr>]]
```

### Examples

```
QUICK Rights .someone.ou1.org
```

```
QUICK Rights User=A* .ou1.ou2.us
```

```
QUICK Rights .someone.ou1.org noone "Common Name"
```

### Notes

The specified objects must exist.

### See also

Equivalencies: List Security Equivalencies  
Groups: Show or Change Group Memberships  
List: List Objects  
Member: Show or Change Group Members  
Show: Show Object and Selected Properties  
➔ p.16 General Options

## 3.28 Security: Run eDirectory Security Check

### Purpose

Examine the whole tree or individual objects for possible security threats.

This check will analyze a list of eDirectory security holes that you might have created during your system setup and report if it detects potential security gaps for your eDirectory tree.

You may restrict and speed up the operation by explicitly requesting one or more security checks on the command line. See the list of performed operations in the Notes section.

### Syntax

```
QUICK Security [<otype>=]<name*> [options]
```

### Examples

```
QUICK Security User=A* IAFB  
QUICK Security * /cont=[Root] /sub
```

### Notes

For the security check to return valid results you need Admin privileges in the checked area of the tree.

The individual security checks examine if ...

- A: [All objects] ... there are excessive rights in the object ACLs
- B: [User] ... passwords are required
- C: [User] ... unique passwords are required
- D: [User] ... grace login limits are set
- E: [User] ... passwords need to be changed regularly
- F: [User] ... passwords need to have a minimum length
- I: [OU, User] ... intruders can and/or have been detected
- J: [User] ... excessive security equivalencies exist
- S: [NCP Servers] ... NLS licenses are used and the bindery context is set
- V: [Volume] ... excessive rights for the root path are assigned

By default, all checks are performed; if you wish to restrict the security checks to one or more operations, just list the respective letters in any order, but without separating blanks.

### See also

Equivalencies: List Security Equivalencies  
Groups: Show or Change Group Memberships  
List: List Objects  
Member: Show or Change Group Members  
Show: Show Object and Selected Properties  
→ p.16 General Options

## 3.29 Server: Execute Command on Server

### Purpose

This option allows you to start server commands on the selected server, or control the mounted server volumes.

As you might guess from the option names, 'LOAD' will load a new NLM, 'UNLOAD' will unload a loaded NLM, 'MOUNT' and 'DISMOUNT' will mount/dismount volumes, and 'NCF' will start an NCF file.

The command prompt will only return after execution of the command, so you should not call commands that require further user input on the console, or the command prompt may not return until that interaction is completed – this is a restriction of the underlying Novell APIs. In batch files, you may also consider preceding the QUICK command with the DOS command "Call" or "Start" (e.g., "CALL QUICK Server ...")

**Syntax**

```
QUICK Server <name> LOAD|UNLOAD|NCF|MOUNT|DISMOUNT <name>
```

**Examples**

```
QUICK Server Srv3 LOAD MONITOR.NLM
QUICK Server myServer NCF SYS:TEMP\myncf.ncf
QUICK Server myServer MOUNT VOL3
```

**Notes**

You need console operator or Admin privileges on the server object to run this command.

You cannot unload an NLM if other NLMs use it; when loading NLMs observe the required NLM load order.

**See also**

→ p.16 General Options

## 3.30 Show: Show Object and Selected Properties

**Purpose**

This option allows you to get basic or extended information on objects and read attribute values.

If you just select one or multiple objects without specifying any attributes, QUICK for eDirectory will display some default attributes for standard objects (Persons, Queues, and Groups).

If you enter '\*' as attribute, QUICK for NDS will retrieve and list all available attributes of the selected objects.

Alternatively you may specify an attribute that you want to have displayed for the objects.

As with the *QUICK Attribute* command, you may concatenate or pipe multiple attributes.

**Syntax**

```
QUICK Show [<type>=]<name*> [<attr>]
```

**Examples**

```
QUICK Show .someone.ou1.org
QUICK Show User=A* "Given Name"
QUICK Show .ou1.ou2.us *
QUICK Show User=* "Given Name;Surname"
QUICK Show User=* "manager|Surname"
```



**Notes**

As usual, you need to enclose objects or attributes in double quotes if they contain blanks.

Be aware that some attributes may be multi-valued (like group memberships or ACLs) or may contain complex information (like login scripts). These will be displayed in multiple subsequent lines as necessary.

Some attributes may contain binary information. These will be displayed as hexadecimal characters.

**See also**

Attribute: View or Change Object Attributes

List: List Objects

Dump: Show All Available Object Attributes

Exists: Check Object Existence

Equivalencies: List Security Equivalencies

Groups: Show or Change Group Memberships

LoginScript: Display Login Scripts

Restrict: Show or Change Object Restrictions

Rights: Get Object's Rights

➔ p.16 General Options

## 3.31 Syntax: Show All QUICK Syntaxes

**Purpose**

List all QUICK for NDS options with their respective syntax and sample calls.

**Syntax**

*QUICK syntax*

**Examples**

*QUICK Syn*

**See also**

Options: List Available Command Line Options

## 3.32 Visible: Change Visibility of Objects

### Purpose

To hide an object or container from the regular user you may set the inherited rights filter for that object. This command line option sets the inheritance rights mask for the object and its attributes, so that only users with Admin rights will be able to view the object and – for containers - its child objects.

You may make the object visible by specifying '+' as last parameter, or make it 'invisible' by specifying '-'.

### Syntax

```
QUICK Visible [<type>=]<name*> +/-
```

### Examples

```
QUICK Visible User=A* -
```

### Notes

You might want to grant someone Admin rights to the object before hiding it. Making containers invisible prevents non-Admin users to see or browse that container and its child objects.

### See also

Delete: Delete Objects  
→ p.16 General Options





## 4 License Agreement

Purchasing QUICK for NDS allows you to use this utility on one tree only. Ask for interesting multi-tree or site licenses, or check our web site at <http://www.WolfgangSchreiber.de> for details.

The full license agreement can be found in the file license.txt – read it carefully before using the software.

On that web site you will also find the latest release of QUICK for NDS and may find update options.

