



# UTILITIES FOR MANAGING NETWORKWARE

## User's Guide

### About JRButils

JRButils v9.0 comprises 122 different utilities. This collection is in five parts:

1. Part A comprises 78 bindery based utilities. These are 16 bit DOS programs and they are available on the Internet in JRB900A.ZIP. These utilities are free.
2. Part B comprises 91 NDS aware utilities for use under DOS. These are 16 bit programs, mostly from Part A , but includes a number of programs which are NDS specific.
3. Part C comprises 32 bit console (character mode) utilities for use from Windows 95/98/ME and Windows NT/2000/XP clients. The 118 utilities include the entire Part B collection, most bindery only utilities from Part A, plus a number of utilities (e.g. for NDPS and GroupWise) for which there are no equivalent 16 bit versions.
4. Part D comprises 32 bit utilities with graphical output. These 98 utilities are the same as in Part B with the exception of *lastlgn*, *licensed*, *requests* and *servtime*, it includes *scanprop* from Part A, *jrbcapt*, *gwdlists*, *nadusers* and the NDPS utilities from Part C, plus *jrblist* which is only in this collection. These programs are still operated from the command line, but they open a window to display their output.
5. Part E comprises 22 utilities which have a full graphical interface for both input and output. With the exception of *jrbpass* which is provided only as a full graphical program, these programs are from Part C.

The programs in JRButils are provided without warranty of any kind. Although, every effort has been made to ensure that the programs are bug free and perform as described, neither the author nor JRB Software Limited accept any liability whatsoever for any damages including lost profits or revenue, or any other consequential damages resulting from the use of these programs comprising JRButils. Any action brought against the author or JRB Software Limited will be under the jurisdiction of the courts of New Zealand.

**For more information use:**

Email: support@jrbsoftware.com  
Fax: NZ +64 3 332 8947  
US (708) 575 1724  
Phone: NZ +64 3 332 5996  
(the least preferred option due to time zone differences from the USA and Europe)  
Web: www.jrbsoftware.com  
Postal: JRB Software Limited  
PO Box 28118  
Christchurch 8000  
New Zealand

Please report any problems. Suggestions for enhancements and new features are always welcome.

**Purchasing JRButils**

The complete collection of JRButils is available on CD at the following pricing:

Software only	\$US550
Software plus documentation	\$US650

A single license for JRButils covers an organisation within one metropolitan area where a metropolitan area is limited to the area contained within a 30 Km diameter circle. Organisations spanning more than one metropolitan area and wanting to use JRButils within each area must purchase more than one copy. A five site license may be obtained for the price of four. For large numbers of sites, please contact JRB Software Ltd.

Every program in JRButils contains in-built help giving basic details on how to use it. The documentation provides much more extensive information including examples. Due to the costs of printing and shipping from New Zealand a manual of nearly 600 pages, it is available only in electronic form. The documentation is provided in Adobe acrobat format which allows it to be viewed and searched via an acrobat reader, viewed via a browser, and it may be printed. There are no restrictions on the number of copies which may be printed for use at the site for which it was purchased.

The purchase price includes all updates to JRButils released within 12 months of purchase. A new version of JRButils is released every December.

---

To continue receiving updates to JRButils after 12 months from the initial purchase, renewal of the registration for JRButils is available at the following pricing, providing that the renewal is paid within 6 months of expiration. After this time, full pricing applies.

Software only	\$US400
Software plus documentation	\$US450

To order JRButils, use either of the on-line order forms at [www.jrbsoftware.com](http://www.jrbsoftware.com) or fill in ORDER.FRM and post it to:

JRB Software Limited  
PO Box 28118  
Christchurch 8000  
New Zealand

Alternatively you may:

Email to: [sales@jrbsoftware.com](mailto:sales@jrbsoftware.com)  
Fax to: NZ +64 3 332 8947  
US (708) 575 1724

---

# Release Notes

## New Programs

### Appadd

Associates users, groups and container objects with ZEN application objects.

### Appdel

Removes associations between users, groups and container objects, and ZEN application objects.

### Gwdlists

Add users to GroupWise distribution lists, removes users from distribution lists and displays the members of lists. It may also be used to change the participation types of existing list members.

### Jblookup

For locating users in NDS and displaying selected attributes specified in a file or chosen from a list. This is essentially a much simplified version of *getname*, but also has the ability to search a particular attribute for a given value e.g. the "Telephone Number" attribute for "7885". It is intended for performing quick look up's on information in NDS. The GUI version can be activated from the system tray.

### Nadusers

Adds users to and removes users from AD forests, domains, organizations and organizational units when using Novell Account Management with Active Directory. *Nadusers* also lists the members of the previously mentioned AD containers.

### Polassoc

Lists the associations of ZEN policy objects. This program has similar display options to *usergrps*.

## Prusers

This displays the queues and operators associated with NDS printers. It is for use with *pradd* and *prdel*.

## Wgrpmemb

For use in a batch file to test if a workstation object is a member of a workstation group. It has similar functionality to *grpmemb*.

## GUI Support

- Created a fully GUI version of the new *jblookup* program.
- Converted *whodidit* to display the output in a listview and this provides considerably greater flexibility for manipulating the results.
- Changed the third party product used to resize all of the component windows when maximizing the main window. The new product does a much better job.
- Numerous minor enhancements documented under individual programs.

## Changes to multiple programs

- Fixed an issue in programs accepting a list of file servers to perform some operation on, where they could produce an application error under certain circumstances in a Pure IP environment.
- Modified those programs accepting a list of file servers to perform some operation on, to also accept a list of servers in a file e.g. *users@s.lis/\**.

---

## Changes to individual programs

### Accexp

- Added /c to the GUI version to force the dialog box to be the topmost window.

### Applist

- Modified to allow the logical operators '&', '~' and '!' to be specified as 'and', 'or' and 'not'. For example ***applist (A and B) or not C*** is equivalent to ***applist (A & B) ~ ! C***.
- Added /m to display the members of groups instead of just the group object names when groups are associated with each application.

### Chkhome

- Added /o to allow the object class to be specified. Possible values are 'user', 'group' and 'template'. This is primarily for use when using wild cards for groups or templates.
- Fixed an issue with locating users corresponding to long directory names when using /c and /d.
- Added /t to force checking that the user has the specified rights (e.g. /t=RWCEMF) to their home directory.

### Creatobj

- Fixed an issue where if you specified a server name before an object name, but not as part of a home directory path, the home directory was not necessarily created on that server e.g. ***creatobj yogi/mandy.sales /d=vol1:users /h*** would create Mandy's home directory on the server to which your current drive was mapped which may or may not be yogi.
- Fixed an issue where it did not set a value for "Grace logins remaining" when a template specified the grace logins allowed. This did not cause problems for the new user, but *nwadmin* and *ConsoleOne* did not display correctly that a grace login limit was set.
- Added support for creating NIMS:ListUser objects. These have a mandatory attribute named "Surname" which is set to equal the object name.

- Updated to work with Novell Account Management 2.1 where an extra step is required for the changes to become visible from NT when adding a user to an NT domain via a template.
- Fixed a problem where it was not granting each user rights to a profile's login script when setting the profile via a template.
- The 32 bit versions can now add users into GroupWise via a template object.

## Deladdr

- Added /b to delete addresses which it does not know how to decode. Novell have not documented all address types, and from time to time, new ones are added.
- Added the ability to recognise type 8 (UDP) and 9 (TCP) network addresses. Note that these differ from Pure IP addresses which are type 1.

## Delobj

- Added /w to remove users from GroupWise as well as NDS.
- Added /\$ to force NDS requests to go to a particular server. This may be necessary when deleting alias objects in a tree containing servers without replicas. If a server without a replica receives the request, it may not refer it to a server holding a replica.

## Domgrps

- Updated to work with Novell Account Management 2.1 where an extra step is required for the changes to become visible from NT.
- Modified to attempt each step of adding or removing a user from a domain group even some have already been done. This allows a partial addition or removal to be completed.

## Domusers

- Updated to work with Novell Account Management 2.1 where an extra step is required for the changes to become visible from NT.
- Added /b for use when adding users to a domain. This adds the domain object to the user's "IWS:Domain Trustees" attribute allowing the account to be administered from NT tools. This setting can also be applied when the user is already a domain member.

- Fixed an issue where it might not remove users from all domain groups when removing the user from the domain.
- Added /v=r to display each domain member's RID (remote ID), when displaying the domain members.

## Dquota

- Fixed a problem where /n=g was not working in the Part D version.
- Added /n=u to display the units beside each value. This does not work with /v due to lack of space across the output line.
- Added /n=f for some European countries to use periods in place of commas and vice versa when displaying numeric values.
- Updated to attempt to determine whether classic directory quotas are enabled for NSS. By default, the server does not return the usage for a directory under NSS unless a quota is set, and *dquota* will then calculate the usage itself if required. When classic directory quotas are enabled, the usage is calculated and returned by the server. Currently, the only means of determining whether classic quotas are enabled is to scan the server 'set' commands which requires console rights to that server.

## Fsupdate

- Fixed a problem which could occur when copying to NSS volumes, where file and directories could be copied to the wrong location. See the detailed description under *netcopy*.
- Fixed another problem which could occur on NW 5 and possibly NW 6 where directories could be copied to the wrong location, again due to one of the NetWare functions behaving incorrectly. This could also cause incorrect behaviour when using /m. The problem is described under *netcopy*.
- Fixed a problem where it sometimes did not process all files when using /f to change attributes.
- Increased the range of attributes which can be set using /a. In addition to the most common attributes which could previously be set, the following can also now be used:
  - c copy inhibit
  - dc don't compress
  - di delete inhibit
  - dm don't migrate
  - ds don't suballocate
  - ic immediate compress



p     purge immediate  
ri    rename inhibit  
t     transactional

- Modified the behaviour of /a and /b (for setting file and directory attributes) so that both '+' and '-' can be used to add and remove attributes in the same command. Previously, it would either add or remove, not both. When neither '+' nor '-' are used, the attributes are set to those specified.
- Added /\$ for use with /d to indicate that a directory has been specified rather than a path and file name. By default, if the lowest level of the path does not exist, it is assumed to be a file name. If /s is also used, *fsupdate* will scan the directory tree looking for files with that name to delete. When /\$ is used, this will not happen.
- Fixed an issue where the NetWare client could time out when an internal copy took more than 3 minutes. This typically occurred only when copying files larger than 100 MB, and has been worked around by doing internal copies in 50MB chunks. Previously, the entire copy was done by sending a single request to the server.
- Modified to allow a comma separated list of file names to be given when copying, deleting and setting attributes e.g. *fsupdate yogi/vol1:apps\\*.exe,\*.dll hewey,dewey,louie/vol1:apps*.
- Modified to copy the modification date and time, and the last access date for directories. Previously NetWare did not store these values for directories.
- Modified to attempt to work around a problem caused by server based anti virus software configured to scan files after they have been written. If the anti virus software opens the file too soon after closure, *fsupdate* cannot set the attributes as this action requires that the file be closed. *fsupdate* now minimizes the delay between closing the file and setting the attributes.

## Fsrights

- Modified to allow multiple servers to be specified and to be able to process all volumes on a server. *Fsrights* now accepts a list of servers as per *fsupdate*, with wild cards allowed and '!' may precede a server name to exclude it. When a path is not specified (e.g. hoki,moki/), all volumes on the server will be processed.

## Getname

- Worked around a problem displaying common names and other names. These are stored in the "CN" attribute with the first value being treated as the common name, the second and subsequent, if present, as the 'other names'. But NDS does not guarantee to return attribute values in a particular order, so an 'other name' could be displayed as the common name and vice versa.

- Modified to display the stream attributes "App:Startup Script" and "App:Shutdown Script" as text attributes rather than binary attributes.
- Fixed a problem where it may not show values for attributes of type "Octet String". However, it showed them correctly when displaying all attributes (/a=\*).
- Fixed an issue where it was not accepting some abbreviated forms of attribute names e.g. "internet" for "Internet Email Address".
- Modified to display the first 100 bytes of attributes of type "Octet String" when using a template. These are displayed in full when using /a=\* and /a=attribute, but were previously ignored for other situations e.g. /a="%login %full %bindery\_property".
- Added /z=o to display only the modification timestamp for attribute values. This option suppresses displaying of the values whereas /z without a value displays both the value and modification timestamp.
- Added the ability to decode network address entries of type 8 (UDP), 9 (TCP) and 13 (URL).
- Added /q to display the attributes given on the command line or in the template file in the format used for displaying all attributes. Any extra text on the command line, or formatting in the template file is ignored.
- Fixed a problem where *getname* could give an application error when using /z to show attribute modification times, and the attribute existed but did not have a value.
- Improved the display of "App:GUID", "GUID", "Other GUID" and "NGW: Visibility" attributes.
- Removed the old function of /f which was an alternative to preceding a file name with '@'.
- Added a new /f where an attribute name can be specified. This is the attribute on which filtering via /n will apply. The attribute does not have to be one being displayed, for example you can filter on location while displaying telephone numbers. When /n is used without /f, filtering will occur on the first attribute displayed other than object's name. Note that filtering does not work on object IDs, creation and modification dates and "Other names".
- Modified to check for the existence of a "City" attribute in the schema. Previously "City" was mapped to "Physical Delivery Office Name", but in eDir 8.5, a "City" attribute was introduced. Now, "City" is mapped to "Physical Delivery Office Name" only when there is no "City" attribute defined.
- Modified the way the creator's and modifier's names are displayed. Previously they were displayed as they are stored i.e. as typed names with commas in place of periods. Now the commas are replaced with periods, and they are displayed like other object names, depending on the value for /y.
- Modified to use a much large buffer for displaying attributes of type "Octet String". It will now display values up to around 60KB in size.

## Getquota

- Modified to place double quotes around paths and object names when producing output for *setquota*. This ensures that *setquota* reads them correctly when they contain spaces.

- Fixed a problem where /w was not working in DOS versions.
- Updated to attempt to determine whether classic directory quotas are enabled for NSS. By default, the server does not return the usage for a directory under NSS unless a quota is set, and *getquota* will then calculate the usage itself if required. When classic directory quotas are enabled, the usage is calculated and returned by the server. Currently, the only means of determining whether classic quotas are enabled is to scan the server 'set' commands which requires console rights to that server.
- Added /x=r to scan the entire tree from [Root] for matching objects. This is achieved in most programs by combining /x and /r, but /r is already used and there are no spare letters.

## Getrest

- Added /b to ignore objects for which the requested attribute does not exist. This option is ignored when displaying all account restrictions.
- Modified the way the creator's and modifier's names are displayed. Previously they were displayed as they are stored i.e. as typed names with commas in place of periods. Now the commas are replaced with periods, and they are displayed like other object names, depending on the value for /y.
- Fixed a problem where filtering on network addresses was not working when a value of 'none' was used.
- Fixed a problem where /u, which forces displaying of user names only, was not working for station restrictions and network addresses.

## Grplist

- Modified to allow the logical operators '&', '~' and '!' to be specified as 'and', 'or' and 'not'. For example *grplist (A and B) or not C* is equivalent to *grplist (A & B) ~ ! C*.

## Jrbcapt

- Modified to set a default value of "LPTx" for the job description field where x is the port number. By default, the NetWare client sets this to "LPT Catch". This setting is now displayed when showing the current capture settings.
- Added /desc to allow a value to be specified for the job description.
- Modified to show the current capture settings when the only parameters given are /local and/or /page. Previously, *jrbcapt* did not do anything when only these parameters were given.
- Fixed a problem where capturing to a queue could result in an "Invalid path" error.

## Jrbcx

- Added /h to allow setting the context in the target tree to that in which the user running the program resides. For example if you are authenticated to the target tree as carol.sales.ny.abc, your context will be set to sales.ny.abc.

## Jrbimprt

- Fixed a problem introduced in JRButils v8.0 where it was not processing volume restrictions in the data file.
- Fixed a problem where it would not create home directories when also setting volume restrictions.
- Added the ability to set NMAS/NFA simple passwords via the control statement "Set simple passwords". The default is not to set these. To set simple passwords, cryptographic services must be installed on the machine on which *jrbimprt* is being run. Further, three NMAS DLLs (nmas.dll, nmasmsg.dll and legacylogin.dll) are required. These are available if the NMAS client has been installed, otherwise they can be copied from *ConsoleOne*'s bin directory on any server where NFA or NMAS has been installed.
- Added the control statement "Set NDS passwords". This allows the setting of NDS for NT, GroupWise, NT Domain and/or simple passwords without setting NDS passwords. The default is to set NDS passwords.
- Fixed an issue where the NDS password could be set after the password expiration time, thereby overwriting the password expiration time (setting the password automatically sets the expiration time to 1/1/1992).
- Fixed an issue where under some circumstances it would fail to detect that a password expiration interval had been set via a template, and having detected that a password expiration date had been set, would set a default value of 40 for the password expiration interval.
- Sometimes when processing large numbers of users (500+) the connection to a server is lost. While this is almost certainly a problem with the NetWare client, *jrbimprt* has been modified to try to detect this at several key points in the program, and to reconnect. However, it is totally impractical to check the validity of a server connection every time, as there are literally hundreds of places in the program where this would need to be done.
- Modified to ensure that if a grace login limit is set, the grace logins remaining are set to the same value, and vice versa. If one is set but not the other, *ConsoleOne* and *nwadmin* may show neither as being set.
- Added support for setting NT domain passwords for existing domain users. The workstation from which *jrbimprt* is being run must be logged into the domain with sufficient privilege.
- Fixed an issue where it was not correctly handling auxiliary classes with mandatory attributes. Values for the mandatory attributes must be set in the same request to the server which adds the auxiliary class. Note that there are still two outstanding problems here. *Jrbimprt* will not work correctly if the control file specifies more than one auxiliary class and two or more of these have identical mandatory attributes. *Jrbimprt* will also not apply an auxiliary class from a

template where the class has mandatory attributes for which the user does not already have values set.

- Fixed an issue where it would fail to change a value for mandatory single valued attributes such as “homeDirectory” for the auxiliary class posixAccount.
- Added a number of new field names which are translated to actual attribute names. Most of these are new attributes in eDirectory (NDS 8.5 or later). They are:

Field Name	Attribute Name
Access card number	accessCardNumber
Assistant phone	assistantPhone
Business category	businessCategory
Car license	carLicense
Cost center	costCenter
Cost center description	costCenterDescription
Country	co
Department number	departmentNumber
Display name	displayName
Employee status	employeeStatus
Employee type	employeeType
First name	Given Name
Home phone	homePhone
Instant messaging ID	instantMessagingID
Is manager	isManager
Mail stop	mailStop
Manager workforce ID	managerWorkforceID
Other phone number	otherPhoneNumber
Personal title	personalTitle
Preferred language	preferredLanguage
Preferred name	preferredName
Prefix	personalTitle
Registered address	registeredAddress
Room number	roomNumber
Site location	siteLocation
Suffix	Generational Qualifier
Toll free phone number	tollFreePhoneNumber
Unique ID	uniqueID
Vehicle information	vehicleInformation
workforceID	workforceID

- For other new attributes with single word names e.g. “Mobile”, “Pager” and “Company”, simply use the attribute name as the field name.

- 
- Modified to allow a tab as the separator character. This must be specified via the control statement "separator=tab" i.e. use the three letters 'tab' rather than a tab character which is treated as white space in the control file.
  - Modified to allow distinguished names with a leading period to appear in the data file. This allows creating or updating users in a different context from that set by the "Name context" control statement. Note that relative names (e.g. john.sales.) are not allowed.
  - Added a "Search NDS" control statement. This may be useful when updating or removing users and the data file is created by a data base which does not know the user's contexts. *Jrbimprt* will search the entire tree for the user. This option is ignored for 'create' mode, and when the user name is a distinguished name with a leading period. An error is given when more than one matching object is found.
  - Added the control statement "Home directory attributes" to allow attributes such as Di (delete inhibit) and P (purge immediate) to be set on the home directory.
  - Fixed a problem where it could cause an application error when encountering a missing integer value in the data file.
  - Modified to allow a time to be specified for "Account has expiration date" and "Date password expires" e.g. "11/12/03 9:00:00". Previously only a date could be specified.
  - Modified to check for the existence of a "City" attribute in the schema. Previously "City" was mapped to "Physical Delivery Office Name", but in eDir 8.5, a "City" attribute was introduced. Now, "City" is mapped to "Physical Delivery Office Name" only when there is no "City" attribute defined.
  - Updated to work with Novell Account Management 2.1 where an extra step is required when adding users to an NDS for NT domain. This update allows the changes to become visible from NT.
  - Added support for GroupWise. Users can be added to or removed from GroupWise, added to or removed from GroupWise distribution lists, and have their GroupWise password, visibility, mailbox expiration date, and description set. Users may be added to GroupWise via a template. The following new control statements have been added:
    - "GroupWise server path" specifies the path to the database.
    - "GroupWise domain object" specifies the name of the domain object.
    - "GroupWise post office" specifies the name of the post office.
    - "GroupWise add users" adds users into GroupWise.
    - "GroupWise remove users" removes users from GroupWise.
    - "GroupWise add distribution lists" gives a list of distribution lists to which each user will be added.
    - "GroupWise remove distribution lists" gives a list of distribution lists from which each user will be removed.
    - "GroupWise distribution lists participation" specifies the participation type (to, cc, bc) for the lists. It may be used when adding members to a list, and to change the participation type for existing members.
    - "GroupWise set passwords" is self explanatory.

“Set GroupWise passwords” is an alternative to the previous statement.

One of the first two statements must be present. In addition the following field names may be used:

“GroupWise description” sets the description.

“GroupWise expiration date” sets the mailbox expiration date.

“GroupWise password” allows a GroupWise specific password to be set.

“GroupWise visibility” sets the visibility.

- Fixed an issue with removing values for “Other names” when “delete property=y” was set.
- Added control statement “Set passwords only on create” for use when import mode = ‘b’ to prevent passwords being set when the object is updated. This may be useful when some users in the data file have already been created, and you do not want to set the password for those users, in case they have already logged in and changed their passwords.
- Fixed a problem where it was not granting each user rights to a profile’s login script when setting the profile via a template.
- Modified the method used to scan the tree for objects with the same name when /c=t is used. The original method used, required more than browse rights to [Root] and so could not be used by local administrators with full rights to only a portion of the tree. Now /c=t works correctly with only browse rights to [Root].
- Modified to require an “alias context” statement when the control statement “create aliases=y” is used. Previously, the name context was used when no “alias context” was specified, which could result in *jrbimprt* trying to create a user and an alias with the same name in the same container.
- Added “Group membership add” and “Group membership remove” control statements allowing adding users to and removing users from groups without having to place the group names in the data file for each user.
- Added a “Create subdirectory” control statement to compliment the existing “Create subdirectories” control statement. This new statement allows a comma separated list comprising a subdirectory name (which may be multi-levelled), a set of attributes to apply to it (e.g. PRiDi), then one or more object name and rights pairs for creation of trustee assignments. For example: “Create subdirectory=data\comp101,diri,jonesk.staff.abc, rwcemf,martinp,rf” will create data\comp101 in each users home directory, apply DiRi attributes to the comp101 directory, then grant jonesk.staff.abc RWCEMF rights and martinp RF rights to the comp101 directory.
- Added the ability to remove directory restrictions by giving a value of ‘none’ e.g. “Home directory restriction=none”.
- Updated to work correctly with extended characters such as Ä, Ö and Ü in object names and attribute values.

## Jrbpass

- Added support for setting NMAS/NFA simple passwords. These can be set in addition to or instead of NDS, and NDS for NT passwords. See further comments under *jrbimprt*.
- Modified to append the user's full name in double quotes to the user name in the edit box after a search is made. This assists in ensuring that you are setting the password for the correct user.
- Added -n=f as a command line option to prevent appending the full name to the user name after a search.
- Modified to make the search button the first tab stop after the user name edit box.
- Added support for setting NT domain passwords when the user is logged into a domain without NDS for NT/Corporate edition/Novell Account Management being present.
- Modified to automatically check the "NDS for NT" checkbox if the user is logged into a domain via NDS for NT.

## Killconn

- Worked around a problem where it was not recognising "not logged in" connections under NW 6. The problem was caused by a change in behaviour of an API under NW 6.

## Listobj

- Fixed a long standing bug where an application error could result from object names longer than 128 characters.
- Fixed another long standing bug where /n failed when the search pattern included spaces.
- Changed /^ to /+ because the '^' character is interpreted by the Windows command line parser whereas '+' is not.
- Modified the way it displays the name of NLS Product Containers. These have a distinguished name of the form "NLS:Product=NetWare 5 Conn SCL+NLS:Publisher=Novell+NLS:Version=510.O=jrb" and when displayed without 'typing' *listobj* displayed them as "NetWare 5 Conn SCL+Novell+510.jrb". However, *nwadmin* and *ConsoleOne* displayed them as "Novell+NetWare 5 Conn SCL+510.jrb" i.e. the publisher appeared first. For unknown reasons, some of the SDK functions used by *listobj*, change the order of the components. *Listobj* has been modified to change them back again. The same applies when displaying license certificate objects within the NLS Product License Container.
- Modified to allow names in files to be enclosed in double quotes.



---

## Lscripts

- Modified to allow a value for /m when copying login scripts from NDS to a mail directory. By default, *lscripts* will copy the script to the user's mail directory on the server in their "Message Server" attribute, but you may now specify the target server via /m e.g. /m=yogi.

## Move\_dir

- Fixed an issue where files and directories could lose their long name when being moved.
- Added the ability to use wild cards. By default it is assumed that files are being moved, but /d has been added to indicate whether to move files, directories or both that match e.g. ***move\_dir j\* yogi/vol1:shared yogi/vol1:mgmt /dd*** will move all directories with the first letter 'j' from the shared directory to the mgmt directory.

## Netcopy

- Fixed a problem introduced in v8.0 where /o may not work. One line of code somehow got lost!
- Fixed a problem which could occur when copying to NSS volumes, where file and directories could be copied to the wrong location. The problem occurred when a directory already existed in the target directory with the same DOS name as the directory being copied. *Netcopy* attempts to change the DOS name of the newly created directory to that of the directory being copied, and the function used for renaming was returning 'success' when it had not renamed the directory. This problem has been worked around, and it has been reported to Novell.
- Fixed another problem which could occur on NW 5 and possibly NW 6 where directories could be copied to the wrong location, again due to one of the NetWare functions behaving incorrectly. The problem occurred when in the same directory, a subdirectory had a long name matching the DOS name of another subdirectory e.g. vol1:users\anne contained directory 'programme' with DOS name 'PROGRA~1', and also contained a directory with long name 'PROGRA~1'. The fault has been reported to Novell.
- Fixed an issue where the NetWare client could time out when an internal copy took more than 3 minutes. This typically occurred only when copying files larger than 100 MB, and has been worked around by doing internal copies in 50MB chunks. Previously, the entire copy was done by sending a single request to the server.
- Modified to copy the modification date and time, and the last access date for directories. Previously NetWare did not store these values for directories.
- Modified to attempt to work around a problem caused by server based anti virus software configured to scan files after they have been written. If the anti virus software opens the file too soon after closure, *netcopy* cannot set the attributes as this action requires that the file be closed. *Netcopy* now minimizes the delay between closing the file and setting the attributes.

---

## Openfile

- Fixed an issue where when using /d=t, *openfile* would stop processing directories after encountering a directory containing no matching files.
- Fixed some minor issues with the display in the fully GUI version when changing between selecting objects, connections or files.
- Fixed a problem in the GUI version where it could sometimes send a user a message when it should not have.

## Pndel

- Added /a to do contextless removal of objects. By default, an object name to be removed gets expanded to a distinguished name, so '\*' for example becomes \*.sales.ny.abc assuming that your current context is sales.ny.abc. Consequently only objects in sales.ny.abc are removed. Use /a to prevent the object name being expanded when no context information. For example *pndel ndps\_1 \* /a* will remove all users regardless of their context.

## Pnsetup

- Added /m=d to force removal of the printer's driver files in addition to removing the printer.

## Profiles

- Modified /d to allow a context to be specified as per /v. By default, when removing users from a profile and wild cards are used, the users of the profile are identified by checking the ACLs of the profile object granting R rights to the "Login Script" attribute. If those ACLs are not present because rights are assigned via a different method, the users will not be found. Now, if you specify a context with /d (e.g. /d=.sales.sf.abc), the *profiles* program will scan all users in and below this context and check if their "Profile" attribute contains the required profile object, and if so remove them from that profile.
- Modified /x to work with adding users to a profile. Previously, it could only be used for scanning for profile objects when displaying the users of each profile.

## Pwdexp

- Added support for setting simple passwords into the GUI version. See further comments under *jrbimprt* on the workstation requirements to set simple passwords.
- Added support for setting NT domain passwords into the GUI version.

- Added /o into the GUI version to replace the "Change it now" and "Change it later" buttons with a single "OK" button when the password is within the expiration period.
- Added /h and /i to the GUI version to control the checking of password check boxes when using /a which prevents the user from selecting the passwords to be set. /h forces checking of the boxes, /i forces unchecking of the boxes. Possible values for both are:
  - n NetWare
  - m NDS for NT, NT domain or local workstation
  - s Simple password

## Quotas

- Fixed a problem where it was not automatically changing the fields displayed at the start of each line, when changing from displaying volume to directory quotas or vice versa, or changing from by user to by directory or vice versa.
- Added support for setting quotas for queues.
- Fixed various minor problems with defaults not changing when switching between setting and displaying quotas, or changing between volume and directory quotas.

## Renobj

- Added /w to 32 bit versions to rename users in GroupWise. By default when /w is used, users will be renamed in both GroupWise and NDS, but /w=g may be used to rename the users only in GroupWise.
- Added /x for use with /w=g to rename the user in GroupWise to their NDS full name.
- Added /h to allow a delay in seconds to be specified. The delay occurs after renaming the user in NDS and before renaming the user in GroupWise. This may be necessary to allow replication of the NDS rename to occur before doing the rename in GroupWise.
- Modified /z to allow an attribute name to be specified. Previously, /z allowed the old object name to be stored as an "other name" i.e. a second or subsequent entry in the "CN" attribute, but /z can now be used to store the old name in any attribute holding a text value e.g. /z=description.

## Requests

- Added /f to display each user's full name in parenthesis after their object name. This may be useful to help identify the person associated with the object name.

## Salvlist

- Added a 'row number' column to the list of possible output columns in the GUI version. This may be useful for example, to count off the number of zero length deleted files in a directory, or the number of files with a given extension such as 'tmp'.
- Fixed a problem in the GUI version where it was not printing the recovered name column.
- Added the recovered name as a selectable column in the GUI version. If this column is not being displayed when a file is recovered, it is added as the right-most column.
- Added a heading and the number of deleted files to the printed output from the fully GUI version.
- Added an option to the GUI version to sort by file or directory extension.
- Fixed an issue where it failed to right justify the first column when it contained sizes or row numbers.
- Added support to the GUI version for saving the contents of the list view to a file when clicking on the 'Save' and 'Save as' File Menu items. Under 'Save as', three options exists: (a) save at the current column widths, (2) save at column widths corresponding to the widest item in each column, (3) save in comma delimited form with each item in double quotes.
- Added the same three options above for printing the contents of the list view to the GUI version.
- Implemented page selection when printing
- Added the following sorting options to the non-GUI versions:
  - by creation date and time
  - by update date and time
  - by last access date
  - by attributes
  - by inherited rights filter
  - by updater
- Modified the fully GUI version to retain its horizontal and vertical scrolling position when a column caption is clicked on to sort the column, and when the "redisplay" button is clicked.

## Schema

- Modified so that when using /o to list the object classes for which an attribute is valid, it indicates if the object class is an auxiliary class.

---

## Setacl

- Added support for setting and removing ACLs for attributes belonging to auxiliary classes.

## Sethome2

- Added support for /g=x. /g is primarily provided as a means of creating home directories when the "Home Directory" attribute is already set but the actual home directories, for whatever reason do not exist. However, allowing /g=x where x means remove any existing trustee assignment, provides a convenient means of removing trustee assignments from existing directories. *Sethome2* will report the home directory already exists and then removes any existing trustee assignment. This could also be achieved via /c=x, but that requires the home directory path be specified, and *sethome2* attempts to create the "Home Directory" attribute, create the home directory, and set/remove a trustee assignment.
- Fixed a problem occurring when using /r to copy paths from the "Home Directory" attribute to the bindery "HOME\_DIR" property where it could attempt to validate the path on the current server rather than the server pointed to by the volume object component of "Home Directory".
- Added /u to specify whether the path component of the "Home Directory" attribute is preceded by a slash e.g. "\users\john" rather than "users\john". Use /u=y to insert a slash, /u=n to omit it. The default is /u=n. /u can be used with /i to change existing entries in the "Home Directory" attribute. One customer reported problems with *ConsoleOne* where paths were not preceded by a slash, although we were unable to reproduce that.
- Changed the existing /o to /v.
- Modified /o to allow the object class to be specified. Possible values are 'user', 'group' and 'template'. This is primarily for use when using wild cards for groups or templates.
- Added /q to query whether or not to process each user.

## Setname

- Removed the ability to process NW 3.x *makeuser* input files to convert full names to lowercase.
- Added support for setting some attributes of type "Path", e.g. the "Path" attribute for directory map objects. A path attribute has three components - an NDS object which is specified as the value on the command line, a path which is specified via /m, and a name space type which can be set via /h.
- Modified /s which replaces a separator in a string value with <CR><LF> pairs. If the specified character is followed by 'c', the separator will be replaced by <CR>'s only, if by 'l' it will be replaced by <LF>'s only. Some attributes e.g. "Novonyx:Proxy List" use one character or the other rather than the <CR><LF> pair.

- Fixed an issue where it was not allowing some truncated attribute names to be used.
- Modified to check for the existence of a "City" attribute in the schema. Previously "City" was mapped to "Physical Delivery Office Name", but in eDir 8.5, a "City" attribute was introduced. Now, "City" is mapped to "Physical Delivery Office Name", only when there is no "City" attribute defined.
- Fixed an issue where it would not work across trees when your current drive was mapped to a server in one tree, and you were trying to modify object attributes in a different tree.
- Added the ability to copy the contents of a file into an attribute of type "Octet String". For example, this allows the copying of a jpg file into a user's "photo" attribute.
- Modified to allow the object name specified via /z to be prefixed with a tree name e.g. /z="manuka/.john.sales.abc". The workstation must have an existing connection to the tree.

## Setowner

- On NSS volumes, changing ownership of files results in the space occupied by the file being immediately added to the new owner's usage when the new owner has a volume quota. If changing the ownership causes the new owner to exceed their quota, the change will fail but no error was reported by the server. *Setowner* now checks that ownership has changed on an NSS volume and reports an error if it has not.
- Modified to work around an NDS 8 bug when searching for objects with underscores in the name when trying to locate an object matching a directory name.

## Setpwd

- Added support into 32 bit versions for setting NMAS/NFA simple passwords via /m=s. These can be set in addition to or instead of NDS and other passwords. See further comments under *jrbimprt*.
- Added support into 32 bit versions for setting AD passwords when using Novell Account Management with W2K or XP. Note that the AD password must match the NDS password (a Novell requirement) which can be set either in the same invocation of *setpwd* or can have been set previously.
- Added /d to specify a delay in seconds before setting an AD password. This may be needed when setting both the NDS and AD passwords because the routines called in PWDNOTFY.DLL check that the AD password being set matches the current NDS password, and it may be necessary to wait for replication to occur. At least some versions of PWDNOTFY.DLL always check the password on the master of the replica so *setpwd* attempts to send an NDS password change to the master when also setting an AD password.
- Removed the existing functionality of /f which indicated a parameter was an input file. This can still be achieved by preceding the file name with '@'.
- Added the ability to force *setpwd* to require the old password in order to set a new one. This is achieved via use of /f.

- Added the ability to reset Intruder lockout before changing a user's password. This is achieved via /h. Sufficient rights are required to the "Locked By Intruder" attribute to change the value, and to the "Login Intruder Attempts" and "Login Intruder Reset Time" attributes to delete their values.
- Added support into 32 bit versions for setting GroupWise passwords via /m=g. These can be set in addition to or instead of NDS, and other passwords.
- Fixed an issue in the GUI version where saving randomly generated passwords to a file was not working correctly.
- Added /v to allow verification of NDS or bindery passwords rather than setting them. It is not possible to verify the other types of password which *setpassword* can set. This feature provides a means of detecting users without passwords set e.g. *setpassword* \* /v/n will report that the password matched on all users in the current context without a password.

## Setquota

- Modified to accept double quotes around paths and object names in an input file to avoid a problem where they contain spaces and the path or name was previously considered to have ended when the first space was encountered.
- Fixed an issue where setting the quota relative to the current usage was not working for volume restrictions.
- Modified to allow setting quotas on queue directories.
- Added /o to specify the object class when the object name contains wild cards.
- Modified to skip volumes named "\_ADMIN" on NW 6 when using /a. *Setquota* already skipped volumes named "NSS\_ADMIN" under NW 5.
- Added /g to allow removal of volume quotas from a volume. The following values may be used:
  - a    remove all quotas
  - n    remove quotas for non-user objects
  - u    remove quotas for unknown objects (the default)
- Added /i to prevent setting a quota when the user or directory already has one.

## Setrest

- Fixed an issue where it was setting GroupWise visibility and mailbox expiration time in NDS only, not in GroupWise so the changes were not effective.
- Fixed an issue where it failed to set maximum concurrent connections on NW 6 due to a bug in NW 6.

## Settrust

- Added /c to force it to validate an object on the server on which the rights are being added or removed. This is primarily for when using *settrust* in a script which creates users via *creatobj* then assigns rights via *settrust*, and avoids an issue where insufficient time has elapsed for replication to occur. *Settrust*, by default, attempts to perform the validation on the server on which the rights are being modified, but does not enforce that because to do so would require that any object being granted rights must exist in a replica on that server which will not always be true. The NetWare client, for reasons known only to itself, may send the validation request to another server unless /c is used.
- Updated to restore inherited rights filters via /b. It can do this for both *settrust* and *rights* commands. No new options are required to do this, *settrust* recognises the type of command in the file.
- Updated to allow double quotes around paths in an input file. Double quotes must be used when the paths contain spaces.
- Fixed an issue where it might not delete all trustee assignments for a file or directory when more than 20 were present.

## Trstlist

- Added /c to control whether file or directory trustee assignments are displayed when displaying trustees for selected objects. Previously, *trstlist* always displayed both file and directory trustee assignments. Note that /d which also allows you to select directories or files, applies only when a path is specified, not an NDS or bindery object.
- Updated to also display and save inherited rights filters when processing files and directories. What is displayed is now controlled via /t which can have the following values:
  - b both trustee assignments and inherited rights filters (IRFs)
  - i inherited rights filters
  - t trustee assignments
- By default, IRFs are displayed only if not equal to [SRWCEMFA]. However, /v can be used to display all IRFs.
- Fixed an issue in the GUI version where it behaved incorrectly when no server was specified.
- Modified the GUI version to produce an error if filtering is enabled but no rights boxes are checked.

## Usergrps

- Added /n=f to display groups one per line with their full names.



---

## Users

- Added a heading and the number of connections to the printed output from the fully GUI version.
- Fixed a problem introduced in v8.0 where it was not showing any connection information unless the user running the program has supervisor or console rights to the server.
- Added the ability to display and sort by the following new fields for each connection:
  - Object class
  - File locks (requires NW 5 or 6)
  - Record locks (requires NW 5 or 6)
  - Server rights (requires NW 5 or 6). These may be Normal, Auditor, Console, Supervisor.
- Added a new column to the fully GUI version for displaying the row number.
- Added support to the GUI version for clearing connections when right clicking in the list view.
- Added support to the GUI version for saving the contents of the list view to a file when clicking on the 'Save' and 'Save as' File Menu items. Under 'Save as', three options exist: (a) save at the current column widths, (2) save at column widths corresponding to the widest item in each column, (3) save in comma delimited form with each item in double quotes.
- Added the same three options above for printing the contents of the list view to the GUI version.
- Implemented page selection when printing.
- Separated the protocol options from the authentication options in the GUI version and added an "All protocols" option as the default.
- Added another checkbox to the authentication options allowing the display of all connections except those which are 'not logged in'.
- Modified so that the output options page displays a format matching that selected on the main options page.
- Added to the GUI version the ability to send a message to a connection.
- Fixed a problem in the GUI version where it would sometimes cause an application error after the exit button has been pressed.
- Fixed in 32 bit versions, an issue where 'NOT-LOGGED-IN' connections were not being displayed on NW 6. This issue existed in all versions due to the NWGetConnectionInformation function indicating an invalid connection number for a 'NOT-LOGGED-IN' connection on NW 6. An alternative function is now used in 32 bit versions, but as it is not available for the DOS platform, DOS versions will not display 'NOT-LOGGED-IN' connections on NW 6.
- Modified the fully GUI version to retain its horizontal and vertical scrolling position when a column caption is clicked on to sort the column, and when the "redisplay" button is clicked.

- Modified to detect connections from Macs using Native File Access and to display their addresses correctly as IP numbers.
- Added /b=m to allow filtering on Macs connecting via Native File Access.

## Wgrpadd

- Added support for specifying '.' as a workstation name in 32 bit versions. *Wgrpadd* will retrieve the current workstation's object name from the registry.

## Wgrpdel

- Added support for specifying '.' as a workstation name in 32 bit versions. *Wgrpdel* will retrieve the current workstation's object name from the registry.

## Wgrplist

- Modified to allow the logical operators '&', '~' and '!' to be specified as 'and', 'or' and 'not'. For example ***wgrplist (A and B) or not C*** is equivalent to ***wgrplist (A & B) ~ ! C***.

## Whodidit

- Fixed a problem where filtering by owner, updater and archiver did not always work when using an input file (it depended on how the object name was given).
- Added /o=# to display the number of files in directories
- Added /o=\$ to display the number of deleted files in directories
- Fixed a problem in the GUI version where once filtering had been used, it was not possible to turn it off without restarting the program.
- Added the ability to sort output by the following fields:
  - owner
  - updater
  - archiver
  - creation date and time
  - update date and time
  - archive date and time

last access date  
logical size  
physical size  
resource fork size (mac files)  
quota (directories only)  
total size (logical + resource fork size)  
attributes  
path length in the name space displayed  
number of files in the directory  
number of deleted files in the directory  
DOS name  
Mac name  
NFS name  
Long name  
FTAM name  
inherited rights filter  
owning name space  
number of extended attribute keys present  
total size of the extended attributes

- Modified to use more efficient functions for retrieving files and directories, so it should process large directory trees a little faster.
- Modified to give an error when no matching files were found and it is processing only a single directory.
- Modified the handling of extended attributes. Now instead of simply displaying whether a file or directory has extended attributes, it displays the number of extended attribute keys for `/o=e`, and `/o=[` has been added to display the total bytes occupied by extended attributes.
- Updated to display update dates and times, and last access dates for directories. NetWare did not previously record these values, and *whodidit* displayed the value as 'None'.

## Programs in JRButils v9.0

### Accexp

*Accexp* is intended for use in a login script where it displays a warning if the user's account is about to expire. The number of days before expiration at which warnings begin is adjustable and defaults to 7. Both text mode and GUI versions are available.

### Appadd

Associates users, groups and containers with application objects.

### Appdel

Removes associations between users, groups and containers, and application objects.

### Applist

Lists the users associated with ZEN application objects. This program has similar functionality to *grplist* and *wgrplist*.

### Bindclos

*Bindclos* may be used on bindery based servers to close the bindery. Use *bindopen* to re-open it afterwards.

### Bindopen

*Bindopen* may be used on bindery based servers to open the bindery. Use *bindclos* to close the bindery first.

## Chkhome

*Chkhome* allows you to check that NDS "Home Directory" attributes and bindery HOME\_DIR properties are set correctly and to display the paths contained therein. Specifically *chkhome* can:

- Display home directories for one or more users. When obtaining the home directory path from the "Home Directory" attribute, the name space in which the path is stored is also displayed.
- Check for each user that the directory stored in the NDS "Home Directory" attribute or in the bindery HOME\_DIR property actually exists, and if it does not, it will scan other volumes on the server for a matching directory.
- For a single directory, or all subdirectories of a directory, check if there is a registered user corresponding to the lowest level directory i.e. for vol1:users\karen, it will check if user Karen exists.
- For each directory, check that the corresponding user's "Home Directory" attribute or bindery HOME\_DIR property contains that directory e.g. for vol1:users\karen it checks that Karen's "Home Directory" contains vol1:users\karen.
- Can display each user's default server.
- Can check the "Home Directory" attribute for lowercase characters in the path. *Nwadmin* allows a path in the DOS name space to be specified in lowercase, but NetWare functions require DOS paths in uppercase.
- Check that each user has the required rights (e.g. RWCEMF) to their home directory.

## Cl\_type

For use in a batch file to determine the type of 16 bit DOS client being used. It returns error levels indicating whether NETX, VLMs or Client32 are loaded.

## Con\_ops

*Con\_ops* allows you to add or remove one or more console operators for a server in a single command. Features include:

- May add or remove user or group objects as console operators.
- May list console operators and optionally sort them.
- Can display only the number of console operators on the server.

## Copy\_obj

Copies NDS objects. New objects may have the same name in a different context, or a different name in the same or a different context. *Copy\_obj* can copy many classes of object, and copies all attributes making changes where necessary. For example, any references to the source object in attribute values such as ACLs are changed to the target object's name.

## Copyuser

*Copyuser* copies users and groups from one NW 3.x server to another retaining all bindery properties and their contents where possible, and copies the contents of mail and home directories, retaining all attributes, trustees, quotas etc as per *netcopy*. *Copyuser* may also be used to replicate users on the same server using an existing user as a template. Other features include:

- Quotas can be adjusted during the copy (for larger block sizes on the destination volume).
- Groups can be substituted.
- Passwords may be left NULL or set to the user name.
- The home directory path may be altered during copying (e.g. from VOL1:USERS to USERS:ECON).
- Groups can be created on the destination server as required when moving users who belong to these groups on the source server.
- Ownership of files can be set to the new user regardless of the original owner.
- Can copy only the contents of home or mail directories. When copying mail directories, it uses the object ID for each user on the source and target servers, and will create the mail directory on the target server if it does not exist. This provides a means of transferring mail directories when the users have been transferred by other means and the object IDs have changed. For example, a combined hardware upgrade and upgrade from NW 3.x to NW 6 can be done by installing NW 3 on the new server, copying the bindery then doing an in-place upgrade to retain the passwords.

## Creatobj

*Creatobj* allows you to create NDS and bindery objects of almost any class or type. It is intended primarily for creating users in batch mode. See also *jrbimprt*.

*Creatobj* can do the following:

- Set a password.
- Create a login script.
- Create a home directory and grant rights to it.

- Store the home directory path in the “Home Directory” attribute.
- Enable password expiration and either expire the password or set it to expire in n days time.
- When creating users, copy attributes from a template.
- Can specify a time delay to occur after creating an NDS user, before attempting to create the home directory and grant rights to it. This gets around an NDS programming problem where a user creation request is directed to one server, the home directory is created on a second but the new user has not yet been replicated to that second server. An NDS request to create an object can be directed to a particular server and *creatobj* will attempt to do this.
- Can initiate a synchronization of the replica immediately after creation of the new object.
- Can add new users into GroupWise from a template.

## Deladdr

*Deladdr* can be used to overcome a problem caused by a bug in NDS where a user’s network address is sometimes not removed from NDS at logout. When a user logs in, the network address of the workstation is added to the user’s “Network Address” attribute. The number of addresses in this attribute is used at login to determine the number of concurrent connections and whether the user has reached their limit. When addresses are not removed, a user may be told they have reached their connection limit when in fact they have not. *Deladdr* can be used to remove addresses where there is no corresponding connection. Because the addresses cannot be linked back to the server to which the user connected, *deladdr* requires a list of servers to be scanned for connections with matching addresses to ensure addresses are not deleted for users currently logged in. *Deladdr* supports the use of IP connections under NW 5 and NW 6, as well as IPX.

## Delattr

Deletes a selected attribute from one or more NDS objects. *Delattr* refuses to delete some attributes where their removal might cause problems. These exclusions include ACLs, “Security Equals”, and partition related attributes. This does not mean that it is safe to use *delattr* to delete all attributes which are not excluded. Many attributes are not set in isolation e.g. removing a user from a group requires modification of two user and two group attributes, and anyone using *delattr* needs to be aware of such relationships. Use *delattr* cautiously, and entirely at your own risk.

## Delobj

*Delobj* can delete almost any type of NDS or bindery object. For safety you may not use wild cards when deleting objects. Nor will *delobj* delete groups with members. Features include:

- When deleting users, *delobj* will delete the user’s home directory if it is stored in the NDS “Home Directory” attribute, or if a path is specified on the command line.

- Can delete each user's mail directory and its contents.
- Can remove users from GroupWise.
- Multiple objects may be deleted using an input file.

## Delprop

Deletes a bindery property for any object on a NW 3.x server. Its primary purpose is to overcome a problem of bindery corruption where *syscon* will report error 234 which means that the link between a property and its data has been lost. *Bindfix* fails to fix this situation which can be resolved by deleting the affected property.

## Disable

Disable disables logins under all versions of NetWare. Use JRButils' *enable* program to subsequently re-enable logins.

## Domgrps

Adds users to, removes users from, and lists the users of Novell Account Management, Corporate Edition or NDS for NT local and global groups. *Domgrps* can also create and delete local and global domain groups, and allows addition of global groups to local groups.

## Domusers

Adds users to, removes users from, and lists the users of NT domains when Novell Account Management, Corporate Edition or NDS for NT is installed.

## Downsrvr

*Downsrvr* may be used by a console operator or supervisor equivalent user to down a NetWare server. By default it will ask for confirmation before downing the server, but you may suppress this.

## Dquota

*Dquota* is intended for use in a login script to display the user's quota, space used and space available. It can handle both volume based and directory based quotas. The format of the output is extremely



flexible, and it is possible to supply a text file with substitution parameters for the quota, space used etc. Warnings may be given if the user's free space falls below a certain threshold.

## Edit\_pjc

Allows the changing of parameters in NDS and bindery print job configurations. Multiple parameters can be changed on each invocation, and they may be changed for selected job definitions or for all definitions, and they may be changed for individual or multiple objects.

## Fsrights

Displays the rights of selected NDS or bindery objects to a single file or directory, files selected via wild cards, or to all the files and/or directories in a directory tree. *Fsrights* allows filtering of the output by rights. For example, the output could be restricted to matching files and directories where the selected objects have the 'S' right, or perhaps have other than 'RF' rights. *Fsrights* can also show how the rights are derived by displaying inherited rights filters and relevant trustee assignments at each level from the volume root down to the selected file or directory.

## Fsupdate

*Fsupdate* allows you to perform file maintenance operations across multiple servers. Its features include:

- Can copy individual files, files selected via wild cards, or entire directory trees to multiple servers. *Fsupdate* attempts to retain all file attributes including ownership, creation and last update dates.
- Can delete individual files, files selected via wild cards, or entire directory trees on multiple file servers.
- Can rename files on multiple servers. This facility may be used to move files from one directory to another providing that the source and target directories are on the same volume.
- Can specify what attributes you want set on copied files and directories.
- Can overwrite, delete or rename files flagged read-only.
- Can copy trustee assignments.
- Can purge files which have been deleted.
- Handles non-DOS name spaces.
- Can retain compression when copying files under NW 4.x onwards.
- Copies sparse files correctly.

- Has a mirroring option to remove non-matching files, directories and trustee assignments in the target directories.

## Getbal

*Getbal* lists account balances, account holds or credit limits for multiple NDS or bindery users.

Features include:

- Can specify a multiplier to change the units of output.
- Can control which values are displayed and in what order.
- The values may be sorted by user name, balance, or credit limit.
- The output can be filtered by value e.g. all users with a balance of less than \$10.00.
- May be used in a login script to display the account balance for each user as they log in.
- Totals are given when displaying balances and holds for multiple users.
- Can list users without credit limits or account balances.
- Can process all users in a container and in all subcontainers of that container.
- Can display balances and credit limits for print servers.
- Can display balances held by individual servers (a bug in replication can result in servers holding different values).
- Can display the default account balance and credit limit given to new users on bindery based servers. This is normally viewed via *syscon*'s "Supervisor Options" - "Default Account Balance/Restrictions".

## Getequiv

*Getequiv* may be used to check for or display security equivalences for NDS objects and bindery users. It can do the following:

- List all security equivalences for one or more users.
- List all users who are security equivalent to another e.g. admin under NDS or supervisor under NW 3.x.
- Check for accounts under NW 3.11 which have been modified using Wolfgang Schreiber's *super* utility allowing supervisor equivalence to be turned on and off at will.
- List bindery users who are managers of their own accounts. Under 3.11, these users can grant themselves supervisor equivalence using *super*.
- List those bindery users with more than 32 security equivalences. NetWare 3.x takes into account only the first 32 when calculating an object's rights.

- In NDS mode check for security equivalence to file server objects. This equates to supervisor equivalence in bindery mode.

## Getname

*Getname* may be used to display a wide range of attribute values for any class of NDS object. The values of single attributes may be displayed, the values for all attributes may be displayed, or a template file may be used allowing the values for multiple attributes to be retrieved and displayed in any format. The template file comprises one or more lines of text with substitution identifiers representing the attributes to be displayed. A non-exhaustive list of substitution identifiers for objects of class user is:

%account_balance	%mailbox_id
%allow_unlimited_credit	%mailbox_location
%city	%message_server (%default_server)
%common_name	%minimum_account_balance
%context	%modification_date
%creation_date	%modifiersname (eDir 8.5 onwards)
%creatorsname (eDir 8.5 onwards)	%network_address
%default_server (%message_server)	%network_address_restriction (%sr)
%department (%ou)	%object_class
%description	%object_id
%email_address	%oid
%email_alias	%other_name
%facsimile_telephone_number (%fax)	%ou (%department)
%full_name	%password_allow_change (%ucp)
%generational_qualifier	%password_expiration_interval (%pei)
%given_name	%password_expiration_time (%ped)
%home_directory	%password_minimum_length (%mpl)
%initials	%password_required (%pr)
%internet_email_address	%password_unique_required (%upr)
%language	%physical_delivery_office_name (%city)
%last_login_time	%postal_address
%last_name (%surname)	%postal_code
%locality	%postal_office_box
%locked_by_intruder (%al)	%profile
%login_disabled (%ad)	%province (%state, %s)
%login_expiration_time (%aed)	%revision
%login_grace_limit (%gla)	%s (%province, %state)
%login_grace_remaining (%glr)	%sa (%street_address)
%login_intruder_address (%bla)	%street_address (%sa)
%login_intruder_attempts (%blc)	%surname (%last_name)
%login_maximum_simultaneous (%mcc)	%telephone_number
%login_name	%title
%login_time (%lld)	%uid

The identifier may be followed by a 'v' and a field width e.g. '%title\10'. A sample template file (t1.dat) is:

```
%login_name,%full_name,%department,%location,%telephone
```

The command **getname \* /r/x/a=@t1.dat /l=all.log** would produce a comma delimited list comprising the user name, full name, department, location and phone number for all users in the NDS tree.

Another template file (t2.dat) is:

```
%login_name
Last Name:      %last_name
Initials:       %init
Given name:     %given_name
Full name:      %full
Other name:     %other
Description:    %desc
Title:         %title
Department:     %depart
Location:       %location
Street::        %street_addr
Postal code:    %postal_code
PO Box:         %post_office
City:          %City
State:         %State
Mailbox ID:     %mailbox_id
Email addr:     %email_addr
Email alias:    %email_al
Language:      %lang
Fax:           %fax
Phone:         %tel
```

The command **getname jane /a=@t2.dat /yd** produces

```
.Jane.mintech.acme
Last Name:      Lomax
Initials:       I.J.
Given name:     Jane
Full name:      Jane I.J. Lomax
Other name:     Jo
Description:    Manager
Title:         Dr
Department:     Research
Location:       K Block
Street:         7777 Long St
Postal code:    54321
PO Box:         PO Box 123
City:          Auckland
State:
Mailbox ID     ABC-123
```

```
Email addr:      :SMTP:Jane@somewhere.co.nz
Email alias:     :jane@somewhere.else.co.nz
Language:       English
Fax:            345-678
Phone:          123-456
```

## Getquota

*Getquota* can display disk quotas, disk usage and disk available for multiple users. It can handle both volume based and directory based quotas under NW 3.11 onwards, and supports both traditional and NSS volumes. Features include:

- Can obtain a user's home directory from the NDS or bindery.
- Can show a user's volume quotas on all volumes or a single volume.
- Can show quotas on all subdirectories of a directory.
- Optional sorting into ascending or descending order of quota, space used, space available, or by user name.
- Optional totals only.
- Can suppress any of the output (quotas, space used, space available).
- Can filter by value e.g. list all users whose usage exceeds 100 MB.
- Output may be in bytes, 4096 byte blocks, KB, MB or GB.
- Can operate on user's mail directories. These can be subdirectories of SYS:MAIL, or PMAIL subdirectories of the user's home directory.
- Can process all users in a container and in all subcontainers of that container.
- Can display all volume quotas on a volume.
- Can display values for queue directories.
- Can accept and display paths in the DOS or LONG name space

## Getrest

*Getrest* displays account restrictions for multiple NDS or bindery users. These include:

- Account is disabled
- Account expired
- Account expiration date and time
- Account is locked
- Address restrictions
- Creation date/time
- Creators name (eDir 8.5 onwards)

---

Force periodic password changes  
Grace logins allowed  
Grace logins remaining  
Intruder lockout bad login address  
Intruder lockout bad login count  
Intruder lockout next reset time  
Last login date and time  
Login time restrictions  
Maximum concurrent connections  
Modification date/time  
Modifiers name (eDir 8.5 onwards)  
Network addresses  
Password is expired  
Password expiration date and time  
Password expiration interval  
Previous login date and time  
Password minimum length  
Password is required  
User can change their own password  
Unique passwords are required  
Proxy password age limit  
Proxy password change allowed  
Proxy password expiration date and time  
Proxy password grace logins  
Proxy password force periodic changes

The features of *getrest* include:

- Displays all restrictions, or a single restriction e.g. minimum password length.
- Can control the order and width of each output field (user name, server/user name, full name, server name, restriction value) when displaying individual restrictions.
- Can sort into ascending or descending order by user name or by restriction value.
- Can filter by restriction value e.g. list all users whose account has expired, or all users whose maximum concurrent connections is not equal to 3.
- Can process users in the specified container and all containers below it.
- Can set an error level indicating the number of matching users. This allows testing in a batch file for example if a particular user's account is disabled.
- Can ignore objects for which the requested attribute does not exist.
- Can display the default account restrictions as set under "Supervisor Options" in *syscon* for bindery based servers.

## Grpadd

*Grpadd* adds one or more users to a group in either NDS or bindery mode. Its features include:

- Can process a single group or a file containing a list of groups.
- Can create the group.
- Can accept one or more users on the command line.
- Can add all members of another group.
- Can add a list of users from a file.
- Can warn if a bindery user has more than 32 security equivalences after being added to the group (NetWare 3.x uses only the first 32 when determining a user's rights).

## Grpdel

*Grpdel* removes one or more users from a group in either NDS or bindery mode. Its features include:

- Can process a single group or a file containing a list of groups.
- Can accept one or more users on the command line.
- Can remove all members of another group.
- Can remove a list of users from a file.
- Can delete the group if it has no remaining members.

## Grplist

*Grplist* lists the members of one or more NDS or bindery groups. Features include:

- Displays the full name of each member.
- Optional sorting by member name.
- Optional totals only.
- Optional listing of all users on the server or in an NDS container.
- Can use logical operators 'and', 'or' and 'not' to list users who are or are not members of a combination of groups.
- Optional listing of members as *grpadd* or *grpdel* commands.

## Grpmemb

*Grpmemb* is intended for use in a batch file to determine if a specific user is a member of a particular group. It functions in NDS or bindery mode and returns an error level.

## Gwdlists

Add users to GroupWise distribution lists, removes users from distribution lists and displays the members of lists. It may also be used to change the participation types of existing list members.

## Home2

This is an NDS version of David Harris' *home* utility for mapping your current drive or a designated drive to a user's home directory or mail directory. *Home2* works in both NDS and bindery mode. In NDS mode it will check first for a "Home Directory" attribute, and if this does not exist, *home2* will check for a bindery HOME\_DIR property. Other features include:

- Can map the drive to the user's mail directory.
- Can map root.
- Can map to a queue's directory.
- Can map to a print server's directory.
- Can search NDS for the specified object.

## Is\_vlms

*Is\_vlms* is intended for use in a DOS batch file to determine whether NETX or the VLMs are in use. It returns an error level.

## Jberrtxt

Accepts a NetWare error code in decimal or hexadecimal, looks it up in JRBERR.MSG (the file of error code translations used by all of JRButils) and displays the description if found. Currently, JRBERR.MSG does not contain descriptions for the error codes returned by print servers, or NDPS.



---

## Jblookup

For locating users in NDS and displaying selected attributes specified in a file or chosen from a list. This is essentially a much simplified version of *getname*, but also has the ability to search a particular attribute for a given value e.g. the "Telephone Number" attribute for "7885". It is intended for performing quick look up's on information in NDS, and can be activated from the system tray.

## Jrbcapt

A replacement for Novell's *capture* program. This incorporates almost all of the functionality of *capture* but has the advantage of allowing capturing of NDPS printers if using the 3.20 or later client under Windows 95/98/ME. The current clients for NT/W2K/XP do not support capturing of NDPS printers.

## Jrbcx

*Jrbcx* is an alternative to Novell's *cx* program without the options to browse the tree. Its primary purpose is to provide a 32 bit program for changing the current context at a command prompt, where execution of a 16 bit program can be very slow. However, it also provides better support than *cx* for multi-tree environments allowing setting of a context in any tree, displaying of contexts for any or all trees and setting a selected tree as the default. It will also establish a connection to another tree if required.

## Jrbimprt

A replacement for Novell's *uimport* for mass creation of users. *Jrbimprt* supports all of *uimport*'s modes of operation i.e. 'c' (create), 'u' (update), 'b' (both create and update) and 'r' (remove) modes. *Jrbimprt* overcomes many of the limitations of *uimport*. Advantages over *uimport* include:

- Supports a wider range of attributes e.g. "Internet Email Address". *Jrbimprt* supports all text attributes and those of type 'integer', 'counter', 'interval' and 'boolean' including those added as custom schema extensions.
- Supports creation of home directories in the LONG name space allowing directories longer than 8 characters to be created. The "Home Directory" attribute will also be set in the LONG name space.
- Supports a "Template" control statement specifying the name of a template object to use e.g. "template=.t\_1.sales.abc". The template may be a user or template object.
- Supports attributes specific to template objects such as "New Object's DS Rights" and "Volume Space Restrictions".

- Supports a "Home directory rights" control statement specifying the rights to be granted to the home directory e.g. "home directory rights=RWCEMF".
- Supports a "Home directory restriction" control statement allowing a restriction to be set on each user's home directory. The restriction is specified in KB e.g. "home directory restriction=5000".
- Supports a "Home directory attributes" control statement allowing attributes such as 'Di' and 'Ri' to be set on the home directory.
- Has an option to check the syntax of the input files and the validity of NDS objects referenced in the data file.
- Has an option to wait for a specified number of seconds after object creation or deletion to allow replication to occur.
- Has an option to initiate a replica synchronization after object creation to speed replication.
- Has an option to standardize the case in new user names. The first character is made uppercase, the remainder lowercase.
- Allows object names to be used as attribute values in the data file to be specified relative to the current context rather than the context in which the new users are to be created. This can be achieved by placing the import control statement "Use current context=y" before the "Name context" statement in the control file.
- Can do two passes through the data file for import modes 'b' and 'c'. Users are created on the first pass, attribute values are set on the second. This potentially avoids replication issues and the need to introduce delays after creating each user. Use the import control statement "Use two passes=y" to enable this option.
- Can create a subdirectory of sys:mail for each user. Use the import control statement "Create mail directory=y" to enable this option. It will be created on the same server as the home directory. By default when deleting users, *jrbimprt* deletes both the home directory and a mail directory (a subdirectory of sys:mail). These actions may be controlled via import control statements. *Jrbimprt* will not delete the home directory if (a) the lowest level directory name does not match the user name, or (b) the user does not have RWCEMFA rights to the directory, but this behaviour can also be changed using import control statements.
- Can create an alias for each new user. This can be done in one of two ways. The first is to specify "alias" as a field name and to give the alias object name in the data file. The name may be a distinguished name with a leading '.', or it may be relative to the context in which the new user is created. The second method is controlled by two import control statements and creates aliases with the same name as the new object:

create aliases=[y/n]

alias context=<context>

- Can create subdirectories of each user's home directory via the "Create subdirectories" import control statement e.g. "create subdirectories=pmail,netscape\ini,photographs". Multiple levels may be created e.g. netscape\ini results in both the netscape and ini directories being created.
- Supports NDS for NT (Corporate Edition). It can add each user to a domain via a template, or via a "domain" import control statement, and can set NDS for NT passwords. Use the import control statement "Set nds for nt passwords=y" to set these in addition to the NetWare passwords. Adding a user to a domain is supported only in 32 bit versions.

- Can set both IPX and IP network address restrictions.
- Does not require a "last name" field. When omitted, the last name is set to the user name.
- It is available in a 32 bit version.
- Supports NDS 8 auxiliary classes via the "Auxiliary classes" control statement.
- Can create groups as required via the "Create groups" control statement.
- Supports a "Target server" control statement. All NDS requests are sent to the designated server to try to overcome replication issues.
- Can set simple passwords used by Novell Modular Authentication Services and Native File Access, via the "Set simple passwords" control statement.
- Can set NT domain passwords via the "Set NT domain passwords" control statement for existing domain accounts when Novell Account Management is not installed.
- Can control whether or not NDS passwords are set via the "Set NDS passwords" control statement. This may be useful if, for example, you want to set only simple passwords.
- Can search NDS for users when it is not known what context they are in via the "Search NDS" control statement. This may be useful when the data file is derived from a database which does not hold context information.
- Supports GroupWise. Users can be added to and removed from GroupWise, added to or removed from distribution lists, and can have their GroupWise passwords, visibility and mailbox expiration time set. These may be achieved by a range of new control statements and field names, or via a template. A GroupWise password may be given separately from the NDS password allowing the two to be different.
- Group membership may be managed via "Group membership add" and "Group membership remove" control statements, rather than using "Group membership" field statements and including the group names in each line of the data file.
- A "Create subdirectory" control statement complements the "Create subdirectories" control statement allowing a single subdirectory of the home directory (possibly multi-level) to be created, the attributes of the lowest level directory to be set, and one or more trustee assignments created, granting objects rights to the lowest level directory. Multiple "Create subdirectory" control statements may be used.

## Jrblist

A graphical utility to display the contents of a text file using the JRButils graphical display routines. This may be useful to redisplay output logged to a file, or to combine output from several programs (e.g. *dquota*, *lastlgn* and *pwdexp* in a login script) and then display it.

## Jrbmap

*Jrbmap* is a replacement for Novell's *map* program, primarily for use in batch files where it will map drives without fuss and piping in 'y's and 'n's. While it does not have all the features of *map*, it has numerous features which *map* does not have. Its features include:

- Can specify alternative paths so that in the event of one being unavailable it will try the next (this can be used for all applications so that in the event of an APPS volume failing, the application will simply be retrieved from the corresponding volume on another server).
- Can save existing mappings in the DOS master environment or the parent environment under NT/2000 and subsequently restore from there.
- Supports 'map root'.
- Supports 'map next'.
- Can delete drive mappings.
- Can map multiple drive letters in a single command.
- Can save all mappings to a file and subsequently restore them.
- Can display current mappings including the paths for local drives.
- Can force truncation of the lowest level directory to eight characters. This could be useful in a login script e.g. **#jrbmap p:=vol1:users\%LOGIN\_NAME /b**
- Can map drives across NDS trees with the username and password specified as options on the command line when there is no existing authenticated connection to the target tree.
- Can map to a user's home directory by specifying the path as ~HOME.
- Accepts on the command line, and displays paths in either the DOS or LONG name space.

## Jrbpass

*Jrbpass* is a graphical utility for changing passwords. Unlike *setpword* which sets them en masse, *jrbpass* allows a user to change their own password, or Help Desk staff to set passwords for individual users. Features include:

- In addition to NetWare passwords, it can set NDS for NT, NT domain, local workstation and simple passwords.
- Can search NDS for a user.
- Administrators changing passwords for other users can optionally retain the current password expiration date, or update it by the password expiration interval.
- Administrators can unlock an account if it is locked by NetWare's intruder detection.

## Jrbpurge

*Jrbpurge* allows selective purging of files in a single directory, in an entire directory tree, or for an entire volume. Features include:

- Supports purging by name.
- Supports purging by size.
- Supports purging by owner.
- Supports purging by deleter.
- Supports purging by deletion date and time.
- Can query whether or not to purge each file.
- Can specify multiple files to purge in a single run e.g. \*.obj,\*.tmp,\*.swp,fred.bak.
- Can purge all volumes on a server in a single command.

## Killconn

*Killconn* clears connections from a file server under all versions of NetWare. Features include:

- Can clear all connections from a server.
- Can clear all not logged in connections.
- Clear connections for objects of a specific class or type.
- Clear connections for objects of a given name.
- Clear specific connection numbers.
- Clear connections which have been logged in for less than or greater than a specified time.
- Clear connections of users who are members of specified groups.
- Clear connections based on their status (bindery, licensed, unlicensed).
- Can accept a list of object names or connection numbers in a file.
- Can query whether or not to clear each connection.
- Can specify a list of objects whose connections are to be retained.
- Has a suicide option where it will logout your connection after clearing all other requested connections.

---

## Killzero

This program will delete zero length files in a single directory or in an entire directory tree. It was written to remove the large numbers of zero length files which in the past could accumulate in student mail directories when using Pegasus Mail with directory quotas enabled. When a user had no space available and an attempt was made to deliver a message, the file was created but could not be written to, leaving a zero length file. Options include:

- Can delete files flagged read-only.
- Can list the zero length files without deleting them.
- Can display only the total zero length files.
- Can purge deleted files (this is recommended as there is no reason to recover an empty file!).
- Can retain certain zero length files by giving a list of one or more file names, optionally containing wild cards.
- Is smart enough not to delete Mac files with a zero length data fork but a non-zero length resource fork.
- Can display paths in the DOS or LONG name space.

*Killzero* can process the home or mail directories for single users, all members of a group, users selected via wild cards, or a list of users in a file. It can also process directories directly.

## Lastlgn

*Lastlgn* can be used in a login script to display the user's previous login date and time. This provides a useful indication of when the account was last used. Note that there is a login script command (lastlogintime) which gives this information, and using this is more efficient than running an external program. However, it has the disadvantage that it is unreliable as the lastlogintime command sometimes retrieves the information from a different server to that on which authentication occurred, in which case it can display not the previous login date and time, but the one before that.

## Licensed

*Licensed* is intended for use in batch files to detect the status of a connection to a file server. It sets an error level indicating whether an NDS connection is licensed or unlicensed, and whether a bindery connection is logged in or not.

## Listobj

*Listobj* lists objects of any or all types or classes across multiple NDS or bindery file servers. Features include:

- Can specify multiple server names, possibly containing wild cards. If a server name is preceded by '!', it is skipped.
- Displays full names and optionally object IDs.
- Can accept a list of names in a file, and will locate them in NDS.
- Optional sorting by user name, object ID or object class.
- Optional totals only.
- Can search the full name field as well as user names.
- Can display full names for users, groups and print servers as *setname* commands.
- Can return an error level if no matching objects are found, providing a means to detect in a batch file if an object of any type exists.
- Can set an error level equal to the number of matching objects.
- Can list any class of NDS object in the nominated container.
- Can search all subordinates of the nominated container.
- Can search from [Root].
- Can search for non-effective classes e.g. device.
- Attaches transparently as GUEST if this account is available on bindery server, or performs a background authentication on NDS servers.
- Can list only permanent or temporary bindery objects under NW 3.x.

## Lscripts

*Lscripts* manages login scripts and print job configurations. It operates under all versions of NetWare. Features include:

- Copies scripts from mail directories into NDS.
- Copies scripts from NDS into mail directories.
- Copies a file into NDS or the mail directories to become the login script.
- Sets NDS scripts to 'include' the mail directory script.
- Creates mail directories for those users who do not have one, and grants the standard trustee rights [RWCEMF].

- Views login scripts for users from NDS or mail directories.
- Can delete login scripts from NDS or mail directories.
- Can view and set login scripts for NDS profile objects, organizations and organizational units.
- Corrects Mac names for mail directories. Older versions of Novell's Install may not change the mail directory names in the Mac namespace during an upgrade (each mail directory is renamed as each user's ID changes).
- Renaming mail directories after restoration of NDS via replication. Object IDs change when restoring so it is necessary to rename mail directories. To do this *lscripts* requires a file containing distinguished names and the object IDs before restoration. *Listobj* can create such a file.
- Renaming mail directories after restoration of NDS from backup. This method assumes that the trustees have been restored, but the directories have not been renamed. Each mail directory is renamed to the ID of the user object which is the subject of the first trustee assignment.
- List mail directories without a matching user.
- Can copy any file into each user's mail directory retaining the file name allowing *lscripts* to be used for tasks such as giving every user the same printcon.dat.
- Can copy print job configurations into mail directories, into NDS, or from one to the other.
- Can display print job configurations.

## Makememb

*Makememb* is an alternative to *grpadd* and *grpdel*. Whereas these allow adding or removing multiple users from a group, *makememb* allows adding or removing individual NDS or bindery users from multiple groups. Features include:

- Can create groups.
- Can specify multiple groups on the command line.
- Can specify files containing groups.
- Can specify a file containing one user name followed by one or more group names on each line.
- Can query whether or not to make each user a member of each group.
- Can warn if a bindery user has more than 32 security equivalences after being added to the group (NetWare 3.x uses only the first 32 when determining a user's rights).



## Mattach

*Mattach* allows you to attach to or detach from multiple servers in a single command. Features include:

- Can specify one or more server names, each of which may contain wild cards, and you may exclude servers by preceding the name with '!' e.g. *mattach rata,toki,p\*,!pn\*/john* would log you into the designated servers as user JOHN.
- Can optionally give the password on the command line.
- Can create unauthenticated and not-logged-in connections.
- Can optionally list all visible servers as per *slist* or *nlist server /b*.
- Can optionally perform a *whoami* function and list all servers to which your workstation is connected.
- Can sort by server name for both the *whoami* and *slist* functions.
- When listing all visible servers, *mattach* can also display the version of NetWare running on each server.
- When listing all visible servers, *mattach* can also display the serial number of each server.
- When listing all visible servers, *mattach* can also display the up time for each server.

## Mgradd

*Mgradd* adds one or more users or groups to a NW 3.x user account manager or to a work group manager. On the command line you may specify multiple entities to become managed by the nominated manager object which may be a user or a group. Each entity may be a user name (possibly containing wild cards), a group name (possibly containing wild cards) or a file containing a list of users or groups. When a group name is specified, either the group object or the members of the group may become managed. The manager may be made a work group manager if desired.

## Mgrdel

*Mgrdel* removes one or more users or groups from a NW 3.x user account manager or from a work group manager. On the command line you may specify multiple entities to removed from management by the nominated manager object which may be a user or a group. Each entity may be a user name (possibly containing wild cards), a group name (possibly containing wild cards) or a file containing a list of users or groups. When a group name is specified, either the group object or the members of the group may become unmanaged. The manager may be removed from being a work group manager if desired.

## Mgrlist

*Mgrlist* lists NW 3.x managers and their managed users. Features include:

- Can list all objects managed by a nominated user or group.
- Can list all managed users on the server.
- Can list all work group managers and/or all user account managers on the server.
- Output may be sorted by object name.
- May output totals only.
- Detects user account managers and work group managers who obtain this status by being security equivalent to another work group manager or user account manager.

## Move\_dir

Moves individual files, or directories and their contents from one location on a volume to another. *Move\_dir* operates by changing the directory entry rather than copying and deleting, so effects the change very quickly.

## Move\_obj

*Move\_obj* moves NDS objects from one container to another. Objects can be moved individually, can be selected using wild cards, all members of a group may be moved, or a list of objects in a file may be moved. When using an input file, the destination context may be given on each line allowing users to be moved to different contexts in a single invocation of *move\_obj*. An alias object can be created pointing to the object in its new location, and a delay may be introduced before creating the alias to allow replication to occur.

## Nadusers

Adds users to and removes users from AD forests, domains, organizations and organizational units when using Novell Account Management with Active Directory. *Nadusers* also lists the members of the previously mentioned AD containers.

## Ndsrghs

*Ndsrghs* evaluates the rights of one or more objects to a target object or to an attribute of the target object. The rights can be displayed for all source objects, or filtering may be applied, and the rights displayed only when they match a specified criteria. *Ndsrghs* can show how the rights are derived.

## Netcopy

*Netcopy* copies individual files or an entire directory tree from one location to another, possibly on a different volume, on a different server, or on a local drive. It preserves the following when copying files and directories:

- All file information including the creation date and time, last update date and time, archive date and time, last access date, inherited rights filters, file attributes, extended attributes, and if possible, file ownership. Assuming the person running *netcopy* has sufficient rights, ownership is retained when the source and target servers are in the same tree, when copying from one tree to another and the owner is in an identical context in both trees, or when copying from one NW 3.x server to another and the owner exists on both servers.
- All directory information including the creation date and time, last update date and time, last access date (if supported), inherited rights filters, directory attributes, extended attributes, and if possible, directory ownership. See the comments above on retaining ownership.
- Optionally, all file and directory trustee assignments.
- Optionally all directory quotas.
- Optionally all volume quotas.

*Netcopy* copies DOS, Long, NFS and Mac name space entries. It will attempt to retain name space information for FTAM, but this has not been tested. Other features include:

- An update option allows you to copy only those files which do not exist in the destination tree, or have more recent update dates. Or, you can copy only those files which do not have an exact match of size and update date in the destination directory.
- Quotas may be removed during copying and restored later. This is often useful when copying to a volume with a larger block size where each file is likely to have more space allocated.
- Quotas may be altered as they are copied.
- Checks that there is sufficient space available before copying a file. This can be suppressed for a slight gain in speed.
- Can optionally ignore hidden files and directories.
- Can copy only those files with the archive bit set with optional resetting of the archive bit on the original and copy of the file.
- Can query whether or not to copy each file.

- Can disable the receipt of broadcasts while copying.
- Copies compressed files without decompressing them.
- Copies sparse files correctly.
- Works across NDS trees.

## Nsrename

*Nsrename* allows the renaming of files and directories in any name space without affecting the name in other name spaces. It can also change the case of names of multiple files or directories and was in fact written to overcome a problem where a particular product failed to function correctly when it encountered NFS names in uppercase.

## NTSetEnv

Takes a text string and an environment variable name as parameters, issues the text string as a prompt, reads the response and assigns it to the environment variable, under Windows NT, 2000 and XP. This is similar to the DOS *readln* utility.

## Openfile

*Openfile* can:

- Display open files by NDS or bindery object name, or connection number.
- Display who has a particular file or files open.

Features include:

- Can display open files for all logged in objects on a server.
- Can display open files for any class of object.
- Can display open files in any name space.
- Can display who has any files open in a particular directory, or in an entire directory tree.
- Can display lock information for open files.
- Can clear connections holding a file open.
- Can set an error level corresponding to the number of open files or the number of users holding a file open.

- Can display a connections' open files in either the DOS or LONG name space, or the name space in which they have been opened.
- Can sort the output as follows:
  - by connection number
  - by directory name
  - by file name
  - by object name
  - data from multiple files, users, or connections can be merged before sorting
  - the sort order can be reversed
- Can display the total number of open files on the server.
- When doing a merged sort by directory, it gives the number of unique files open.

*Openfile* can also work in conjunction with an NLM allowing users without console operator status to display open files and to clear connections. Usage of the NLM is controlled by group membership.

## Orgroles

*Orgroles* adds occupants to, removes occupants from and lists the occupants of organizational roles. Wild cards are supported allowing processing of multiple organizational roles in a single command.

## Pnadd

Adds users, operators and managers to NDPS printers.

## Pndel

Removes users, operators and managers from NDPS printers.

## Pndeljob

Deletes jobs enqueued to NDPS printer agents

---

## Pnlist

*Pnlist* lists the jobs enqueued to NDPS printer agents. Features include:

- Wild cards may be used.
- The default output gives the job position, job ID, user, description, size, date/time submitted and status for each job.
- You may design you own output selecting from 9 fields and choosing the size and order of each field. The fields are:
  - Description
  - Job entry time
  - Number of documents in the job
  - Job ID
  - Job position
  - Job priority
  - The job status (pending, paused, held, etc)
  - The name of the user who enqueued the job
  - Number of bytes to be printed
- Output may be limited to jobs submitted by a particular user.
- Output may be filtered on job description.
- The number of jobs can be returned as an error level.

## Pnsetup

Installs NDPS printers to a workstation, removes installed printers, sets the default printer, lists installed printers, and can list the installable printers in a context.

## Pnusers

Lists users, operators and managers of NDPS printers.

## Polassoc

Lists the associations of ZEN policy objects. This program has similar options to *usergrps*.

## Pqadd

*Pqadd* adds one or more users or groups as operators or users of the specified print queue. Features include:

- Can use wild cards in the queue name.
- Can use an input file containing queue names.
- Can also add print servers to a queue.
- You may specify multiple entities on the command line, each of which may be a single user, one or more users selected via wild cards (e.g. z\*), a group, or a list of user names in a file.
- Can make a group object a queue user or operator, or the individual members of the group may be made queue users or operators.
- Can add both queue users and operators (but not print servers) in a single invocation.

## Pqdel

*Pqdel* removes one or more users or groups from being operators or users of the specified print queue. Features include:

- Can use wild cards in the queue name.
- Can use an input file containing queue names.
- Can remove print servers from a queue.
- You may specify multiple entities on the command line, each of which may be a single user, one or more users selected via wild cards (e.g. z\*), a group, or a list of user names in a file.
- Can remove a group object from being a queue user or operator, or the individual members of the group may be removed as queue users or operators.
- Can remove both queue users and operators (but not print servers) in a single invocation.

## Pqdeljob

*Pqdeljob* deletes one or more jobs from a print queue. Features include:

- You may specify one or more job numbers on the command line, '\*' meaning all jobs in the queue, a range e.g. jobs 10-20, or user names possibly including wild cards.
- Can chose to have *pqdeljob* ask for confirmation before deleting each job.
- Can delete jobs by banner name.

## Pqlist

*Pqlist* lists the jobs in one or more NDS or bindery queues on one or more file servers. Features include:

- Wild cards may be used in the queue name.
- Any number of bindery file server names may be specified (e.g. hoki,moki,toki/laser\*) or you may specify 'A' meaning all servers to which you are currently attached. File servers preceded by '!' will be skipped.
- The default output gives the job number, user, description or file name, size, date/time submitted and status for each job.
- You may design you own output selecting from 18 fields and choosing the size and order of each field. The fields are:

Banner user name

Banner file name

Connection number of the workstation from which the job was submitted

Description

Job entry time

File name (may be blank if the job was captured directly from an application)

Job number

Form number

Form name

Job position in the queue

Number of copies

The path associated with the file being printed

The name of the job file in the queue directory

Target server name if specified

The job status (open, ready, user hold, etc)

The name of the object which enqueued the job

Job execution time

Number of bytes to be printed

- Output may be limited to jobs submitted by a particular user.
- Output may be filtered on banner user name.
- The number of jobs can be returned as an error level.



## Pqmovdir

Allows the changing of directories for an existing print queue. It may be changed to a different directory, a different volume, or under NDS to a different server.

## Pqmovjob

*Pqmovjob* allows print jobs to be copied from one NDS or bindery queue to another possibly on a different server. The jobs may optionally be deleted from the source server.

## Pqreljob

*Pqreljob* allows jobs to be released from user and/or operator hold on NDS or bindery queues. It may optionally be used to place existing jobs on user or operator hold.

## Pqstatus

*Pqstatus* allows you to view and alter the status of multiple print queues across multiple servers. It displays:

- Whether jobs can be submitted to the queue.
- Whether print servers can attach to the queue.
- Whether print servers can service the queue.
- The number of jobs in the queue.
- Attached print servers and their connection numbers.

Features include:

- Can use wild cards in queue names.
- Any number of bindery file server names may be specified (e.g. hoki,moki,toki/laser\*) or you may specify 'A' meaning all servers to which you are currently attached. File servers preceded by '!' will be skipped.
- Allows you to toggle or set to a specific value: (a) jobs can be submitted to the queue (b) servers can attach to the queue (c) servers can service the queue

---

## Pqusers

*Pqusers* lists the queue users, operators and/or print servers for one or more queues. Features include:

- Wild cards may be used in the queue name.
- Any number of bindery file server names may be specified (e.g. hoki,moki,toki/laser\*) or you may specify 'A' meaning all servers to which you are currently attached. File servers preceded by '!' will be skipped.
- The list of users, operators or print servers for each queue may be sorted.
- Can limit the output to the total number of users, operators and/or print servers for each queue.
- User or operator groups may be expanded so that all user objects which are queue users or queue operators are listed.
- Can output *pqadd* commands for users, operators and print servers.

## Pradd

*Pradd* works in NDS mode only allowing queues to be added to NDS printers. The default queue may be set, and priorities may be assigned to queues when adding them.

## Prdel

*Prdel* works in NDS mode only allowing queues to be removed from NDS printers. The default queue may be removed, but NetWare will automatically make the first of the 'other' queues the default.

## Profiles

The *profiles* program adds profiles to users, replaces profiles for users or removes profiles for users. It can process a single user, or multiple selected via wild cards, group membership, or a list in a file. *Profiles* can create a profile object, but currently cannot assign a login script to it, but you can use *lscripts* to do this. It can also display the users of a particular profile either by scanning the objects which have rights to the profile script (this is quick), or by scanning the NDS tree (this may be slow).

---

## Prusers

This displays the queues and operators associated with NDS printers. It is for use with *pradd* and *prdel*.

## Psadd

*Psadd* adds one or more users or groups as operators or users of the specified print server. Features include:

- Can use wild cards in the print server name.
- Can use an input file containing print server names.
- Can create the print server(s).
- You may specify multiple entities on the command line, each of which may be a single user, one or more users selected via wild cards (e.g. *z\**), a group, or a list of user names in a file.
- Can make a group object a print server user or operator, or the individual members of the group may be made print server users or operators.
- Can add both print server users and operators in a single invocation.
- Under NDS, *psadd* can add printers to an NDS print server.

## Psdel

*Psdel* removes one or more users or groups from being operators or users of the specified print server. Features include:

- Can use wild cards in the print server name.
- Can use an input file containing print server names.
- You may specify multiple entities on the command line, each of which may be a single user, one or more users selected via wild cards (e.g. *z\**), a group, or a list of user names in a file.
- Can remove a group object from being a print server user or operator, or the individual members of the group may be removed from being print server users or operators.
- Can remove both print server users and operators in a single invocation.
- Under NDS, *psdel* can remove printers from an NDS print server.

---

## Psusers

*Psusers* lists NDS and bindery print server users and operators for one or more print servers on one or more file servers. Features include:

- Wild cards may be used in the print server name.
- Any number of bindery file server names may be specified (e.g. hoki,moki,toki/laser\*) or you may specify 'A' meaning all servers to which you are currently attached. File servers preceded by '!' will be skipped.
- The list of users and operators for each print server may be sorted.
- Can limit the output to the total number of users and operators for each print server.
- User or operator groups may be expanded so that all user objects which are print server users or operators are listed.
- Can output *psadd* commands for users and operators.
- Can list the queues serviced by print servers.
- Can list the printers associated with a print server, and optionally the printer numbers.

## Pwdexp

*Pwdexp* is intended for use in a login script on NDS or bindery servers, where it displays a warning if the user's password is about to expire. The number of days before expiration at which warnings begin is adjustable and defaults to 7. Both text mode and GUI versions are available.

*Pwdexp* can also force a password change before or after password expiration. It will prompt for and change the password. However, there is a bug in all versions of Novell's login programs where it still says the password has expired and asks if you want to change it, even when the password has already been changed by a program run within the login script.

## Renobj

*Renobj* allows renaming of NDS or bindery objects of any class or type. Features include:

- When renaming a user, it will check for the existence of an NDS "Home Directory" attribute or bindery HOME\_DIR property. If either is found, *renobj* will rename the user's home directory to match the user's new name and update the contents of "Home Directory" or HOME\_DIR.
- Can specify the home directory path on the command line when the user does not have a "Home Directory" attribute or HOME\_DIR property.
- Can process a file containing one old name and one new name per line.

- Can change the case of the names of existing NDS objects to all lowercase, all uppercase, or the first letter uppercase and the remainder lowercase.
- Can create an alias with the original name of the object and pointing to the renamed object.
- Can rename users in GroupWise.

## Requests

*Requests* continuously displays the total requests, bytes read and bytes written for up to 20 connections since the program began running. This allows easy identification of workstations imposing the greatest load on a server. The output is sorted into descending order by whichever column you select, and the column on which you are sorting can be changed without restarting the program. *Requests* can perform a background authentication to the nominated server and works correctly across NDS trees. Both text mode and GUI versions are available. NW 4.x or later is required.

## Salvlist

*Salvlist* lists deleted files in a single directory or for all directories in a tree. Features include:

- Can display any combination of up to 12 properties of each file e.g. size, owner, deleter etc.
- Output may be sorted by any field.
- Can search for files with a particular name by owner, deleter, file size, and deletion date and time.
- Can display only the number of deleted files in each directory.
- Files may be recovered based on any selection criteria, with automatic renaming if a file with the same name.
- Files can be moved to a different directory on the same volume upon recovery.
- Can request that only the most recently deleted file with a particular name be recovered.
- Can display and recover deleted directories on NSS volumes.
- A fully GUI version is available.

## Schema

Displays information from the NDS schema. The following may be displayed:

- Object classes in the schema. Wild cards may be used to list only a subset of the defined classes.
- For each object class, the names of attributes which are valid for that class.
- For each object class, full details of attributes which are valid for that class.
- A list of attributes defined in the schema. Wild cards may be used to list only a subset of defined attributes.
- A list of attributes with the object classes for which the attribute may be used.
- The *schema* program may be used before and after a product install to identify changes made to the schema by the installation.

## Serv\_cmd

Allows various server console commands to be executed from a work station under all versions of NetWare except NW 3.x. These are:

- Add a name space to a volume
- Mount a volume
- Dismount a volume
- Execute an NCF file
- Load an NLM
- Unload an NLM
- Display the loaded modules
- Display the values for "set" commands
- Display file cache information
- Display directory cache information
- Display cpu information
- Display general information
- Display the amount of memory and memory allocation details
- Display NCP statistics
- Display link support layer (lsl) information
- Display known networks
- Display known servers
- Display login and TTS status
- Down the server
- Disable logins
- Enable logins
- Issue "set" commands
- Manage a server's bindery contexts

Some commands require supervisory rights, others require console operator status. This version can perform a bindery login to the nominated server, or perform a background authentication.

## Servtime

*Servtime* displays the time on one or more NW 3.x or NDS servers, and allows the time to be set on 3.x servers.

## Setacl

*Setacl* can set, delete and view ACLs for a single object, multiple objects selected using wild cards, all members of a group, or for a list of objects in a file. Its features include:

- Can process both object ACLs and attribute ACLs.
- Can process ACLs for any object class or all object classes.
- Allows you to add or delete en masse, objects' rights to one of its own properties. For example, it can grant each member of a group read access to his/her own login script.
- When viewing ACLs, all ACLs for each object can be displayed (this is the default).
- Can limit the ACLs displayed to those for a particular attribute or trustee, or those containing particular rights.
- Can search the entire tree from [Root]. This together with the ability to filter by attribute, trustee and rights allows tasks to be performed such as listing all ACLs in the tree for which a particular object is a trustee, or to list all ACLs granting the supervisory right.
- Supports the "Password Management" pseudo attribute.
- Supports attribute inheritance.
- Supports NDS 8 auxiliary classes.
- Can save ACLs in a form allowing them to be restored, also by *setacl*.

## Setbal

*Setbal* sets account balances or credit limits for a single NDS or bindery user, each member of a group, one or more users selected using wild cards, or a list of users in a file. Features include:

- Both the account balance and credit limit may be set to a specified value, or they may be adjusted by a given amount (e.g. add \$10 to each member of group ABC).

- When using an input file, the balances may be in the file allowing a different balance to be set for each user.
- Can delete accounting holds. This is not normally required but a bug in an older version of A.N.D. Technologies' *Pcounter* could result in large random account holds remaining in existence.
- Can set balances and credit limits for print servers.
- Can process objects in a nominated container and in all containers below that container.
- Can set the account balance or credit limit to the default value used when creating accounts on NW 3.x servers. This is normally set under "Supervisor Options" - "Default Account Balance Restrictions" in *syscon*.

## Setcx

*Setcx* scans an NDS tree for a given object, and if found, sets the default context to that in which the object was found. It functions with only browse rights for [Public] to [Root] which is granted by default at installation. Features include:

- Can prompt for the object to search for.
- Can specify the text for the prompt.
- Can locate objects of any class.
- Can begin the search at any specified location in the tree.
- Can store the distinguished name of the object in the environment.
- *Setcx* will work from a NW 3.x server provided an NDS tree can be found. Where a preferred tree is not set for the client, a tree can be specified on the command line, but this feature cannot be used to override any existing connection to a tree.

## Setequiv

*Setequiv* allows you to make one or more objects security equivalent to another. On bindery servers, *syscon* will run out of memory when you attempt to do this when there are around 3,000 or more users in the bindery. *Setequiv* can also check for and restore the security of bindery user accounts modified by Wolfgang Schreiber's *super* utility (NW 3.11 and earlier). It allows input of object names from a file.



## Sethome2

*Sethome2* stores users' home directories in their "Home Directory" attribute or bindery HOME\_DIR property. Features include:

- When a root directory is specified, (e.g. moa/vol1:users), *sethome2* will automatically append the user name to obtain the home directory path for each user.
- An absolute path can be specified when the home directory does not match the user name.
- Can create the home directory if it does not exist. The user will be assigned [RWCEMFA] rights to the created directory by default but you may specify alternative rights.
- Can modify the rights to existing home directories.
- Can delete "Home Directory" attributes and HOME\_DIR properties.
- Can set or delete each user's default server.
- Can copy the home directory path from the bindery HOME\_DIR property to the NDS "Home Directory" attribute.
- Can copy the home directory path from the NDS "Home Directory" attribute to the bindery HOME\_DIR property.
- Allows the home directory to be stored in the "Home Directory" attribute in any name space.
- Can store paths using '/' or '\' as separators of path components.
- Can convert '/' to '\' and vice versa for paths already stored in the "Home Directory" attribute. Some applications making use of this attribute fail to function when they encounter '/'.
- Can create home directories without modifying the contents of the "Home Directory" attribute or HOME\_DIR property.
- Can control whether the path component stored in the "Home Directory" attribute is preceded by '/' or '\' e.g. "\users\john" versus "users\john".
- Can add or remove leading '/' or '\' characters from the path component of existing values in the "Home Directory" attribute. We have received reports of *ConsoleOne* requiring a leading '\' but have been unable to reproduce this.

## Setname

*Setname* was originally written to set text attributes (hence its name) but its functionality has been extended over time to allow setting of a range of other types of attributes. Its current repertoire is:

- Text attributes such as full name, given name, generational qualifier, description, department, city and email address. *Setname* may also be used to change the case of existing values for text attributes.

- Attributes holding object names as values.
- Attributes using the "Integer" syntax e.g. "Login Intruder Limit".
- Attributes using the "Counter" syntax e.g. "Login Grace Remaining".
- Attributes using the "Interval" syntax e.g. "Intruder Lockout Reset Interval".
- Boolean attributes e.g. "Detect Intruder" and "Lockout After Detection".
- Attributes using the "Typed name" syntax e.g. "App:associations". This syntax comprises three fields - an object name, a level and an interval.
- Attributes using the "Facsimile Telephone Number" syntax e.g. "Facsimile Telephone Number".
- Attributes using the "Email Address" syntax e.g. "Email Address".
- Attributes using the "Object Class" syntax allowing NDS 8 auxiliary classes to be applied to objects.
- Attributes using the "Network Address" syntax e.g. "Network Address Restriction".
- Attributes using the "Back Link" syntax such as those used by NDS for NT.
- Attributes using the "Path" syntax e.g. the "Path" attribute for Directory Map objects.
- Attributes using the "Octet String" syntax e.g. "photo".

*Setname* supports multi-valued attributes allowing addition and deletion of individual values. In bindery mode, *setname* can set the full name for users, groups and print servers.

Features of *setname* include:

- Can set the same NDS attribute value for multiple objects.
- Can add one or more attribute values for multi-value attributes.
- Can remove individual attribute values for multi-value attributes.
- Can convert existing attribute values to entirely upper case, to entirely lowercase, or to a mixture of upper and lowercase.
- Can use an input file containing object names and attribute values.
- Can copy an attribute value from another object.
- Supports setting multi-line values for attributes such as "Description".

## Setowner

*Setowner* can change the ownership of individual files, selected files in a directory tree, all files and directories in a directory tree, or can treat each subdirectory as a separate tree and set ownership to the user corresponding to the subdirectory name e.g. for vol1:users\hewey set ownership to hewey, for vol1:users\dewey set ownership to dewey etc. Other features include:

- Can obtain a user's home directory from the NDS "Home Directory" attribute or the bindery HOME\_DIR property.
- Allows you to specify the name of the new owner.
- Can set ownership to objects of any class.
- Can change ownership of only those files without a valid owner.
- Can process user's mail directories.
- Can display paths in the DOS or LONG name space.
- Can set the updater for files.
- Can set the archiver for files.

## Setpassword

*Setpassword* can set passwords for multiple objects. It operates in either NDS or bindery mode, and its features include:

- Can set simple passwords.
- Can set GroupWise passwords.
- Can set NDS for NT passwords.
- Can set AD Domain passwords when Novell Account Management is installed.
- Can set NT Domain passwords.
- Can specify multiple bindery server names, optionally containing wild cards. Names preceded by '!' are skipped.
- Can set passwords for a range of users e.g. comp001-comp100.
- Can set null passwords.
- Can set the password to match the user name.
- When using an input file, it is possible to set a different password for each user.
- Can generate random passwords of any length. These may optionally be written to a file, along with the user name and optionally the server name and user's full name. The random passwords can be generated without them actually being set.
- Can be used by anyone, but if you are not supervisor or equivalent or a manager of the target object, the old password is required.
- When changing passwords for other users, NetWare by default expires the password if a password expiration interval is set. *Setpassword* can retain the current expiration date, or update it by the password expiration interval.
- Can verify passwords i.e. determine if a given password is the user's current password.

- Can identify users without passwords.
- Can reset intruder lockout before setting a user's password.

## Setquota

*Setquota* sets quotas for multiple users. It handles both volume based and directory based quotas under NW 3.11 onwards, and supports both traditional and NSS volumes. Features include:

- Can obtain user's home directories from the NDS "Home Directory" attribute or the bindery HOME\_DIR property.
- Can set a specific quota, or increment/decrement a quota by a given amount e.g. 2 MB, or by a given percentage e.g. 20%.
- Quotas may be specified in units of bytes, 4096 byte blocks, KB, MB or GB.
- Can set quotas on user's mail directories. These may be the subdirectories of SYS:MAIL or the PMAIL subdirectories of the user's home directory.
- Can process all subdirectories of a directory, or all directories in a tree.
- Can query whether or not to set a quota for each user or directory
- Can process objects in a nominated container and in all containers below that container.
- Quotas may be removed.
- Can remove all quotas from a volume, or all quotas for which no corresponding users exist.

## Setrest

*Setrest* sets account restrictions for multiple NDS or bindery users. These include:

- Account is disabled
- Account expiration date and time
- Account is locked (NW 3.x only)
- Address restrictions
- Force periodic password changes
- Grace logins allowed
- Grace logins remaining
- GroupWise mailbox expiration
- GroupWise visibility
- Last login date and time
- Login time restrictions
- Maximum concurrent connections
- Password minimum length
- Password expiration date and time
- Password expiration interval
- Password is required
- Previous login date and time (NW 3.x only)

Proxy password age limit  
Proxy password change allowed  
Proxy password expiration date and time  
Proxy password grace logins  
Proxy password force periodic changes  
Unique passwords are required  
User can change their own password

*Setrest* may also set the default account restrictions displayed under "Supervisor options" in *syscon* on bindery based servers.

The following points should be noted under NDS:

- NDS allows station restrictions to be set for protocols other than IPX. *Setrest* currently sets only IPX and IP restrictions.
- *Setrest* can process users in the specified container and all containers below it.

## Settrust

*Settrust* performs many of the functions of Novell's *rights* program. This includes setting or removing and displaying trustee assignments, setting and displaying inherited rights filters, and displaying the rights of the person running *settrust* to selected directories and files. Under NW 4.x onwards it can display the rights of any object to selected directories and files. *Settrust's* advantages over Novell's *rights* program include the ability to specify and display paths in the long name space, the ability to set or remove trustee assignments for multiple objects in a single command, the ability to display rights for another user and the existence of a 32 bit version.

## Setvdate

Changes the creation date for a NetWare volume. This may be useful to overcome a problem with Apple Macintosh machines which distinguish volumes by their creation dates and times. If two NetWare volumes have exact same creation dates and times, a Macintosh client cannot distinguish between them.

## Sparse

*Sparse* identifies sparse files. These are files with 'holes' in them where no data exists, and so NetWare has not allocated disk blocks for these parts of the file. Many sparse files are created accidentally by misbehaving applications, and appear as large files when listed in *explorer*. However, they can have as little as one block of disk space allocated. These files can cause problems when copied, or backed up. If the program reading the sparse file reads it as an ordinary file, NetWare returns zero values for the

missing data, and if the data is written to tape or disk, the new file will have all blocks allocated. This can cause problems such as:

- Copies failing because a directory quota or volume quota has been reached.
- User's being over quota if the application transfers quotas after copying the files.
- Running out of space on a volume.
- Slow backups as multi-megabytes of zeroed data is transferred to tape.

*Sparse* has the following features:

- Can scan all files in a directory tree.
- Can scan all volumes on a server.
- Can display the block bit map for each sparse file.
- Can specify a minimum size for files to be checked.
- Reports both the physical size (bytes of disk used) and logical size (reported by 'dir') for each sparse file.
- Can restore the last access date on files which are checked. Because it has to open the file to determine if it is sparse, the last access date is updated by default.
- Can skip compressed files as these are decompressed when checked.
- Displays paths in the DOS or LONG name space.

## Trstlist

*Trstlist* display trustee assignments for multiple NDS or bindery objects, or for selected directories or files. Features include:

- Can display trustees for files, directories, all directories in a tree, or for an entire volume.
- Can display trustees for any class of object on any or all volumes.
- Can display inherited rights filters.
- Optional sorting by user, directory or rights.
- Optional filtering by trustee rights e.g. *trstlist* can list only those trustees with supervisory rights.
- Can list the trustee rights as *settrust*, *rights* or *grant* commands allowing them to be saved in a form in which they can be restored via a batch file.
- Can optionally list the trustee rights as *settrust*, *rights* or *remove* commands allowing them to be saved in a form in which they can be removed via a batch file.
- Can optionally list trustee rights for an object which are inherited through security equivalence. As a user is security equivalent to groups belonged to, this includes trustee rights

inherited via group membership. Under NDS this includes trustees of each of the container objects forming the object's distinguished name, plus [Public] and [Root]. Every object is security equivalent to each of its parent objects.

- Can display only those files and/or directories without trustees.
- Can display trustee assignments for all volume objects in an NDS tree.
- Displays paths in the DOS or LONG name space.

## Usergrps

*Usergrps* lists the groups to which one or more NDS or bindery users belong. Features include:

- Optional sorting by user name.
- Optional sorting of the groups belonged to.
- Can exclude selected groups e.g. everyone.
- Can process users in the specified container and all containers below it.
- Flexible output formatting options.

## Usermgrs

*Usermgrs* lists the managers of users or groups on NW 3.x. You may list the managers for a single user, multiple users selected via wild cards, all members of a group, for one or more group objects, or for a list of users and/or groups in a file. Features include:

- Output may be sorted by object name.
- The managers of each object may be sorted.
- Can exclude certain managers from being listed.

## Users

*Users* displays information about logged in objects, and not-logged-in connections under all versions of NetWare. It has the following features:

- Can list logged in objects across multiple servers. Wild cards may be used in server names, and servers may be excluded by preceding the name with '!'.
- Will attempt to attach transparently to other servers as GUEST, or in NDS mode it will perform a background authentication.

- Can display not logged in connections.
- Can select from 17 possible output fields and control the order of each field and its width. These fields are:
  - Combined network and station address
  - Connection number
  - Connection status (licensed, unlicensed, bindery). This requires console operator status.
  - File locks
  - Full name of the logged in object
  - Location of the workstation
  - Login date
  - Login time
  - Network address
  - Object class
  - Object ID
  - Object name
  - Period for which the object has been logged in
  - Record locks
  - Server name
  - Server rights (normal, auditor, console, supervisor)
  - Workstation address
- Can display where a user is logged in. This requires a text database mapping network addresses to locations. This file may also contain wild card entries which are used when no exact match is found. This feature allows, for example, entries for different networks.
- Can sort the output on any of the fields.
- Can filter output on numerous fields.
- Can display the total users logged in on each server.
- Can show members of selected groups who are logged in.
- Can set an error level equal to the number of matching logged in objects.
- Can display connections for Z.E.N. workstation objects.
- Can display connections based on the connection status (bindery, licensed, unlicensed).
- Can display connections based on whether they are using the IPX or TCP/IP protocols, or AFP via native file access.



## Vol\_info

*Vol\_info* displays information for a single volume, volumes selected via wild cards, all volumes on a server, volumes across multiple servers, or volumes selected via the volume objects in NDS. The information includes:

- Name spaces loaded.
- Volume size, space used, space available and percentage full.
- Free (purged) blocks, purgeable blocks and not yet purgeable blocks.
- Number of directory entries allocated, used and available.
- All information available from the 'volumes' option in *servman* or *monitor* (NW 4.x onwards).
- Can produce a list of volumes on each server.

*Vol\_info* can also produce summaries of space usage, disk block usage and directory entries for selected volumes or all volumes on a server.

## Wgrpadd

*Wgrpadd* is for use with workstation objects and workstation groups which are added to the NDS schema when Z.E.N.works is installed. *Wgrpadd* is a modified version of *grpadd* and has similar functionality allowing a single workstation object to be added to a workstation group, all members of another workstation group to be added, or a list of workstation objects in a file to be added.

## Wgrpdel

*Wgrpdel* is for use with workstation objects and workstation groups which are added to the NDS schema when Z.E.N.works is installed. *Wgrpdel* is a modified version of *grpdel* and has similar functionality allowing a single workstation object to be removed from a workstation group, all members of another workstation group to be removed, or a list of workstation objects in a file to be removed.

## Wgrplist

*Wgrplist* is for use with workstation objects and workstation groups which are added to the NDS schema when Z.E.N.works is installed. *Wgrplist* is a modified version of *grplist* and has similar functionality allowing listing of members of one or more workstation groups, and allowing the use of logical operators 'and', 'or' and 'not' to form an expression combining group names. This program

can also display the user histories for workstation objects and the date and time that each entry was added.

## Wgrpmemb

For use in a batch file to test if a workstation object is a member of a workstation group. It has similar functionality to *grpmemb*.

## Whodidit

*Whodidit* displays selected information about files and directories which is useful in determining when they were created, modified, last accessed, archived and by whom. It can list any combination of the following fields:

- Archiver - the object which last archived the file
- Archive date and time
- Attributes
- Creation date and time
- DOS name
- Extended attributes – number of
- Extended attributes – total size
- FTAM name
- Inherited rights filter
- Last access date
- Logical size as shown by a 'dir' command or Explorer
- Long name
- Mac name
- NFS name
- Owning name space
- Owner
- Physical size (for compressed or sparse files)
- Quota (for directories)
- Resource fork size for Mac files
- Update date and time
- Updater - the object which last updated the file

The user may select which of the above are to be displayed, in what order, and the width of the columns. Alternatively, a template file containing text and substitution identifiers may be given. This allows any combination of values to be given in any format and may be useful to create a batch file of commands incorporating file names. The output can be filtered on any value e.g. it is possible to display details of files greater than a certain size, files which are compressed, or all files and directories owned by a particular name space.

*Whodidit* has the ability to use a template file to format the output. The file can contain one or more lines of text and substitution identifiers representing informational items, the volume, path or file name, or any combination of these in any name space. This allows commands to be built for files or

directories matching some filter criteria. For example, *whodidit* could scan a volume for files not accessed during the last 6 months and via the template file, a batch file could be created to copy these to some other location and then delete them.

*Whodidit* can output owner, updater and archiver information in a form suitable for input to *setowner*. This option may be useful if NDS is to be removed from a server and subsequently reinstated. All object IDs will change when NDS is reinstated with loss of this information which is stored as object IDs in the directory entry tables. Saving this information with *whodidit* allows it to be restored via *setowner*. This facility may also be used to restore ownership for files and directories copied using *copy*, *xcopy* and *ncopy*.

## Notes and Known Limitations

### NDS Issues

- A feature of NDS is that in general, it is not necessary to know on which file servers an object exists. A server receiving a request for information about an object which is not in a replica held on that server, will contact other servers to obtain the information. The programming interface to NDS was designed on the assumption that requests do not need to be directed to specific servers. But this has caused a problem during object creation where the creation request can be sent to one server, and a subsequent request to add an attribute to the new object can be sent to another. The second request will fail when insufficient time has elapsed for replication to occur. The server receiving the request will recognise that it holds a replica in which the object should exist, therefore it will not contact other servers, and returns an error corresponding to "no such object". Novell have provided a work around to this problem which is implemented in JRButils. Any NDS object name given on the command line or in an input file may be preceded by a file server name e.g. moki/.barney.sales.xyz. Each program will attempt to direct all NDS requests to that server. A batch file being used to create objects should make use of this e.g.

```
creatobj moki/.%1.admin.xyz
grpadd moki/.dialin.admin.xyz .%1.admin.xyz
setacl moki/.%1.admin.xyz full_name [root] r
```

- It is not possible to use NDS tree names. Object names and paths may be preceded by file server names, but not by tree names. However, it is possible to change the default tree using *jrbcx* prior to running other programs such as *listobj*. The ability to use either a tree or a server name is expected to be provided in the next release of JRButils.
- When sorting by object name, and typed names are being displayed, the type is included in the sorting i.e. all objects beginning 'cn=' will precede all objects beginning 'o=' and 'ou='.
- In some programs, a parameter may comprise a server name and an NDS object name e.g. karo/.john.devpt.jb. In this situation where a distinguished name with a leading period is given, '/' rather than '\' must be used to separate the server and object name. The reason is that '\' is used to 'escape' certain characters including '.' and '=' to indicate that they form part of the object's common name. Therefore karo\john.devpt.jb represents a valid object with a common or relative distinguished name of karo.john. In all other situations, '/' and '\' can be used interchangeably to separate server and object names.
- NDS allows use of '/' in object names causing a further complication. While koro/john is a valid object name, by default programs in JRButils will treat this as both a server name and an object name. A '/' in an object name may be 'escaped' by '\' to prevent it being treated as a separator between a server and object name. Only the first occurrence of '/' need be 'escaped'. A server name cannot include '/' so 'bill\bob/bazza' must represent an object named bill/bob/bazza.
- NDS allows the use of '\*' in object names, yet this character also functions as a wild card. Not surprisingly, Novell recommend that this character not be used. JRButils support the use of '\*'

in object names but make no attempt to distinguish it from a wild card. Consequently, where wild cards are valid, it is treated as a wild card which may result in the operation being performed on more objects than intended. Note that while '?' is a valid wild card on bindery servers, it is not treated as a wild card under NDS.

- Due to a bug in the client SDK, partial names with a trailing period may not be expanded correctly.
- Due to another bug in the client SDK, names may be truncated incorrectly when two containers, one immediately below the other have the same name. JRButils uses canonical names internally. If the current context is test.jrb and a program was to display the name john.test.test.jrb relative to the current context, it would be output as 'john' instead of 'john.test'.
- This release of JRButils has limited support for Z.E.N. Works related objects. Novell have yet to release to developers, the information required to develop programs for managing these objects.
- This release of JRButils works correctly with NDS v8.x with one exception but the utilities have not been updated to incorporate specific NDS v8.x support. The NDS v8.x functions in the SDK are still under development, and at the time of writing (September 2001) are incomplete. For example, it is not possible to scan an entire tree for selected objects, unless one server in the tree holds replicas of all partitions. The one known problem with JRButils is due to an NDS v8 bug where the function NWDSSearch fails to return any objects when the search pattern includes an underscore. For example *getrest s\_\* lld* returns no matching objects even when, for example, s\_jones exists in the current context. The bug was acknowledged by Novell in July 2000 and has been fixed in eDirectory 8.5 but it appears that the fix will not be incorporated into NDS 8.

## File System Issues

- While JRButils accept paths on the command line in both the DOS and LONG name spaces, they work with DOS paths internally as there are only a limited range of functions in the client SDK which can accept a path in a non-DOS name space. Each program will first check the validity of a path in the LONG name space when the LONG name space is loaded on the host volume. When that fails the validity in the DOS name space is checked. A limitation exists when a path in the LONG name space includes a file name incorporating wild cards. A file name without wild cards can be converted to the corresponding DOS name but one with wild cards cannot. As the programs work with DOS names internally, a name containing wild cards must be a DOS name. For example, vol1:photography\hippo\* is valid, vol1:photography\hippopotamus23 is valid, but vol1:photography\hippopotamus\* is not as a search for hippopotamus\* in the DOS name space will fail.
- Novell's August 1998 document entitled "Troubleshooting High Utilization for Novell NetWare 4.1x" records an issue with an SDK function named NWScanObjectTrusteePaths which is used to scan for trustee assignments belonging to both bindery and NDS objects. When the volume being scanned has more than 2.5 to 3 million directory entries, the search process scanning the directory entry table may lock the volume until the search is completed. This excludes any other activity on the volume such as reading and writing from taking place.

The high utilization caused by the search process together with the volume lock may cause users to lose connections, logins to be very slow, and in extreme cases the server may hang or abend. This problem is limited to NW 4.11 onwards, as earlier versions had a maximum of 2 million directory entries per volume. Programs in JRButils potentially affected by this problem are `trstlist`, `settrust`, `copy_obj`, and `copyuser`. Currently Novell do not have a solution to this problem other than to use NSS volumes under NW 5 and 6.

- A bug in some versions of NFS.NAM (the server loadable module implementing NFS name space support) for NW 4.11 and 5.0 will cause the server to crash when `nsrename` is used to rename a file in only the NFS name space. Read the section describing `nsrename` for more information.
- NSS volumes behave differently from traditional volumes with respect to extended characters in file names. Extended characters are not allowed in DOS file names under NSS and will be removed from the name. This may lead to unexpected changes in DOS file names when files are copied to an NSS volume. On traditional volumes, some extended characters will differ between name spaces. For example if a Mac file is created containing the character represented by ASCII value 0xF9, the character 0xC4 will appear in its place in the long name. On NSS volumes, the 0xF9 character will also appear in the long name. These differences in behaviour may cause problems with applications using affected files.
- Setting large numbers of directory quotas via `setquota` on NSS volumes when the directories contain many files may lock the server. This problem may be avoided by changing the default value for the NSS `numWorkToDos` parameter at the server console e.g.  
`nss/numWorkToDos=100`.
- On traditional file system volumes, NetWare keeps track of the total space occupied by files in every directory and its subdirectories. On NSS volumes, NetWare does this only when a directory quota is set. This is a significant disadvantage as without a quota set, the only way to determine the usage is to sum the space used by every file in the directory and its subdirectories, and this may be time consuming where there are large numbers of files and subdirectories. `Getquota`, `dquota` and `whodidit` are affected by this. They will sum the space used when no quota is set.
- The total reads and writes by connection displayed by the *users* and *requests* programs may be incorrect. Due to a bug in NetWare under NW 5.0, NW 5.1 prior to SP5, and NW 6 without a support pack installed, the total reads and writes apply to traditional volumes only. The values displayed do not include reads and writes from NSS volumes. However, this problem was fixed in SP1 for NW 6.0, and SP5 for NW 5.1.
- *Netcopy* and *fsupdate* may fail to set file attributes on servers with anti virus software installed and which is configured to scan files after they have been written. Setting attributes requires that the file is closed, but the anti virus software may re-open it after *netcopy* and *fsupdate* have closed it and before they have set the attributes. Both have been updated to minimize the delay between closure and setting the attributes. Macintosh files are even more susceptible to this problem as there may be a resource fork to copy, and Macintosh specific fields such as the finder information must be copied separately from the DOS attributes.

## General Issues

- Programs cannot make changes to the environment of the command prompt in which they are running under Windows 95/98/ME. This affects the *jrbmap*, *setcx* and *ntsetenv* programs.
- Programs such as *deladdr* and *users*, which process a list of server names, may appear to hang when an invalid server name is given, and the client is configured to use Novell's Service Location Protocol (SLP), but no Discovery Agent (DA) is present on the network. They will in fact time out from waiting for a response from a DA after around 15 seconds. The solution is to turn off SLP at the client (under the NetWare client's advanced settings), or to enable a Discovery Agent.
- The *lscripts* and *edit\_pjc* programs cannot perform any actions requiring access to bindery print job configurations stored in mail directories (subdirectories of sys:mail). This problem was reported to Novell in 1998, they have acknowledged the fault, but it does not appear that it will ever be fixed.
- Reports have been received of 32 bit programs reporting error 0x8872 (invalid owner) when used from NT workstation v4.0 with NetWare client v4.60 and earlier. This problem has occurred in various programs when accessing or modifying NDS. It occurs under a range of versions of the NetWare client for NT, and appears randomly and inconsistently. It appears to have been fixed in the v4.70 and later clients.
- When running on an NT workstation client, some programs in JRButils (both 16 and 32 bit versions) may fail to execute and report that "the directory handle table is full". This problem arises because an application is allocating directory handles then failing to release them. Each connection to a NetWare server has an associated connection handle table that can hold up to 255 entries. Normally a few handles are allocated to permanent drive mappings and the rest are available for temporary use by applications. This problem is not caused by JRButils. It has been experienced by organizations which do not use JRButils when attempting to run Novell's programs such as *auditcon* in sys:public. At the time of writing, the offending application is unknown. The problem can be worked around by logging out and logging in again.
- If a program is producing output and control-c is pressed under NT 4.0 with service pack 6 installed, or under Windows 2000, a blue screen of death may result, or the machine may just reboot. This is a Windows bug. Pausing the output before pressing control-c seems to avoid the problem.
- Windows NT and 2000 allow multiple commands to be given on a line with '&' being used as the command separator. This treatment of '&' as a command delimiter affects a number of programs in JRButils which make use of '&'. *Grplist* and *wgrplist* use '&' as an 'and' operator, and other programs such as *trstlist* allow '&' to precede a group name to indicate that the action should be performed on the group's members and not the group object. To prevent '&' being intercepted by the windows command interpreter, it must be enclosed in double quotes e.g. *trstlist "&group"* or *grplist g1 "&" g2*.
- All programs accept substitution identifiers denoted by '%' characters in log and error file names, and some programs such as *dquota* and *getname* make other uses of '%' characters on the command line. When these programs are run from a batch file, the batch command interpreter assumes that '%' characters are present to denote environmental variables whose

name should be replaced by the variable's value. To prevent interpretation of the '%' characters use two consecutively e.g. *getname* \* /a="%%login\_name\30 %%telephone"

## Issues in DOS Versions

- Novell no longer support the 16 bit environment, and the DOS utilities are built using the last version of the 16 bit development kit released in 1999. There are a couple of known issues:
  1. The SDK function NWParsePath which is used in all programs accepting a path as a parameter, appears to fail when the workstation has two CDROM drives. The programs work perfectly when MSCDEX is not loaded, or the second CDROM drive is disconnected.
  2. A report was received of *grpmemb* working correctly on some workstations but not others. An older version built using SDK V6 worked correctly on all machines. The reason for this was not determined, but whatever the problem is, it may affect other programs.
- The 16 bit NDS aware programs are large, typically 300 KB or greater. The size results from the amount of NDS specific code linked in from the Client SDK libraries. Much of this is believed to be related to authentication. The two main consequences of having large executable files are that they are slower to load, and those that do sorting have considerably less memory available for holding information to be sorted. Where NDS support is not required in the DOS environment, the utilities without NDS support in Part A should be used where possible. The 32 bit versions in Parts C and D are smaller because they make use of dynamic link libraries (DLLs), and these should be used when using Windows 95/98/ME and Windows NT/2000 clients.
- The 16 bit NDS aware programs must be able to read the unicode files appropriate to the workstation's country and code page settings when operating in NDS mode. For each country and code page combination, four files are required. For example, when the country is the US and the code page is 437, the files are:

UNI\_437.001

437\_UNI.001

UNI\_COL.001

UNI\_MON.001

The unicode files are placed in the nls subdirectories of sys:login and sys:public when NetWare is installed. However, neither of these directories is searched by default. The search sequence to locate the unicode files is:

1. The directory from which the program was run.
2. An nls subdirectory of the directory from which the program was run.
3. The search paths.



---

The programs cannot perform NDS operations when the unicode files matching the workstation's country and code page are not found. Searching for the unicode files is incorporated into the client SDK which is used by JRButils to interface to NetWare. Where the programs in JRButils search for the unicode files is therefore beyond the control of the author.

- An SDK bug manifests itself in the 16 bit NDS aware versions of *getrest* and *getname*. When processing large numbers of NDS objects, using the VLM 1.21 client, they will run out of memory after 200-300 objects. This memory leak is specific to version 1.21 of the VLMs; it does not occur with v1.20 or with Client32.
- Programs such as *deladdr* and *users* which process a list of file server names, may not be able to locate NW 5 and 6 servers running pure IP. Novell have released new SDK functions for finding servers in a pure IP environment, but they were released after support for the 16 bit SDK ceased. The only solution is to use the 32 bit versions of these utilities.
- The *users* and *killconn* programs cannot identify NOT-LOGGED-IN connections on NW 6 due to a bug in the function call used under NW 6. The 32 bit versions of these programs have been updated to use a newer function which behaves correctly but which is not available for 16 bit programs.
- Some programs may run very slowly under Client32 V2.7 for DOS and Client32 v3.0 for Win95/98, yet run normally under other client versions. This is due to a client problem which can be solved by replacing the file IPX.NLM in the workstation's directory holding the Client32 files, with a more recent version from the NetWare client V3.02 for Win95/98. Novell's Technical Information Document 2945214 gives more information.
- The maximum length of any command under DOS is 128 characters, and the computer will beep when you attempt to enter a longer command. Under windows, commands can range up to 2048 characters. However, when running a 16 bit program, windows allows commands longer than 128 characters to be entered but will then fail to run the program reporting "The system cannot execute the specified program".
- All programs in JRButils providing an option to sort the output, do so by building linked lists in memory, then sorting in memory. The 16 bit DOS programs in Parts A and B are limited in the amount of data they can hold for sorting which is constrained by the available memory below the 640 KB boundary. The 32 bit versions have no such limitation, and should be used whenever possible, particularly when sorting large numbers of items. In addition, the 32 bit versions incorporate a very much faster sorting algorithm. It has been observed though that when producing sorted lists of large numbers of NDS objects, the retrieval of object names from NDS takes an order of magnitude longer than the sorting.

---

## Incompatibilities with Previous Versions

Every attempt is made to ensure compatibility with earlier versions of JRButils but occasionally there are good reasons to make changes. In V9.0 the following changes should be noted.

- The function of */f* in *getname* has changed. Previously this was an alternative to preceding an input file name with '@'. Now, it allows an attribute to be specified for which filtering via */n* is applied.
- In *listobj*, */^* has been changed to */+* because the ^ character is used by the Windows command line parser, and so was stripped from the command unless it was enclosed in double quotes.
- The function of */o* in *sethome2* has changed. Previously it indicated that the input file contained home directory paths as well as object names, but this has now been assigned to */v*. Now, */o* may be used to indicate the object class for <entity>. This is consistent with most other programs which use */o* for this purpose.
- The function of */m* in *setname* has changed. Previously it indicated that the input file was a NW 3.x *makeuser* file and that *setname* should convert the full names to mixed case. This functionality is no longer available. Now, */m* is used to specify the "Path" value when setting attributes using the "Path" syntax.
- The function of */f* in *setpword* has changed. Previously this was an alternative to preceding an input file name with '@'. Now, it may be used to force *setpword* to require the old NDS password before setting a new one.
- In programs such as *getname* and *setname*, "city" could be used as an alternative name for the "Physical Delivery Office Name" attribute. However, an attribute named "city" was introduced in eDirectory 8.5. Now, any program given "city" as an attribute name will check for the existence of this in the schema. When the attribute is defined, it is used instead of mapping "city" to "Physical Delivery Office Name".

# Which Program to use

## File system

### Operation

### Program

Deleted files (and directories on NSS volumes)

Listing

by creation date/time  
by deletion date/time  
by deleter  
by name  
by owner  
by size

*Salvlist*  
*Salvlist*  
*Salvlist*  
*Salvlist*  
*Salvlist*  
*Salvlist*

Purging

by deletion date/time  
by deleter *Jrbpurge*  
by name  
by owner  
by size

*Jrbpurge*  
  
*Jrbpurge*  
*Jrbpurge*  
*Jrbpurge*  
*Salvlist*

Salvaging

Directories

Copying retaining all attributes

*Netcopy*

Deleting

*Fsupdate*

Displaying information

any object's rights to

*Fsrights*

attributes

*Settrust*

creation date and time

*Whodidit*

disk usage in

*Whodidit*

expanded (uncompressed) usage in

*Getquota*

inherited rights filters

*Getquota*

*Whodidit*

*Settrust*

*Trstlist*

names in non-DOS name spaces

*Whodidit*

owner

*Whodidit*

owning name space

*Whodidit*

*Move\_dir*

Moving

Quotas - see below

Renaming in the DOS name space

*Fsupdate*

Renaming in all name spaces

*Nsrename*

Renaming in any name space

*Nsrename*

Setting attributes

*Fsupdate*

Setting inherited rights filters

*Settrust*

---

Files	Setting ownership	<i>Setowner</i>
	Compressed – identifying	<i>Whodidit</i>
	Copying retaining all attributes	<i>Netcopy</i>
	Deleting	<i>Delobj</i>
		<i>Fsupdate</i>
	Deleting zero length files only	<i>Killzero</i>
	Displaying information	
	any object's rights to	<i>Fsrights</i>
		<i>Settrust</i>
	archiver	<i>Whodidit</i>
	archive date and time	<i>Whodidit</i>
	attributes	<i>Whodidit</i>
	creation date and time	<i>Whodidit</i>
	extended attributes	<i>Whodidit</i>
	inherited rights filter	<i>Whodidit</i>
		<i>Settrust</i>
	last access date	<i>Whodidit</i>
	logical size	<i>Whodidit</i>
	moving	<i>Move_dir</i>
	names in non-DOS name spaces	<i>Whodidit</i>
	owner	<i>Whodidit</i>
	owning name space	<i>Whodidit</i>
	physical size (data fork only)	<i>Whodidit</i>
	resource fork size (Mac files)	<i>Whodidit</i>
	total size (data + resource fork)	<i>Whodidit</i>
	update date and time	<i>Whodidit</i>
	updater	<i>Whodidit</i>
	Open files - see below	
	Renaming in the DOS name space	<i>Fsupdate</i>
	Renaming in any name space	<i>Nsrename</i>
	Renaming in any name space	<i>Nsrename</i>
	Setting archiver	<i>Setowner</i>
	Setting attributes	<i>Fsupdate</i>
	Setting inherited rights filters	<i>Settrust</i>
	Setting ownership	<i>Setowner</i>
	Setting updater	<i>Setowner</i>
	Sparse files - see below	
Mail directories (SYS:MAIL)	Copying	<i>Netcopy</i>
	Copying to another server and changing ID	<i>Copyuser</i>
	Creating	<i>Copyuser</i>
		<i>Creatobj</i>
		<i>Jrbimprt</i>
		<i>Lscripts</i>
	Deleting	<i>Delobj</i>
		<i>Jrbimprt</i>
	Displaying quotas for	<i>Getquota</i>
	Identifying ownerless directories	<i>Lscripts</i>
	Renaming	<i>Lscripts</i>

---

Mapping a drive to	<i>Home2</i>
Setting quotas for	<i>Setquota</i>
Setting ownership	<i>Setowner</i>
Open files	
Listing	
by connection	<i>Openfile</i>
by NLMs on the file server	<i>Openfile</i>
by user	<i>Openfile</i>
	<i>Requests - GUI</i>
all open files on server	<i>Openfile</i>
in a particular name space	<i>Openfile</i>
locks held	<i>Openfile</i>
Clearing connections holding a file open	<i>Openfile</i>
Quotas	
Directory quotas	
copying with the directory	<i>Netcopy</i>
	<i>Fsupdate</i>
displaying for multiple users	<i>Getquota</i>
displaying in a login script	<i>Dquota</i>
setting, removing, adjusting	<i>Setquota</i>
Volume quotas	
copying to another volume	<i>Netcopy</i>
displaying all quotas on a volume	<i>Getquota</i>
displaying for multiple users	<i>Getquota</i>
displaying in a login script	<i>Dquota</i>
setting, removing, adjusting	<i>Setquota</i>
Sparse files	
Identifying	<i>Sparse</i>
Displaying the block bitmap	<i>Sparse</i>
Trustee assignments	
Copying	
by saving as <i>rights</i> , <i>grant</i> or <i>settrust</i> commands	<i>Trstlist</i>
with files and directories	<i>Copyuser</i>
	<i>Fsupdate</i>
	<i>Netcopy</i>
Deleting	
by saving as <i>rights</i> , <i>remove</i> or <i>settrust</i> commands	<i>Trstlist</i>
directly	<i>Settrust</i>
for objects which no longer exist	<i>Settrust</i>
Home directories	
modifying	<i>Sethome2</i>
setting	<i>Sethome2</i>
Listing	
by directory or file	<i>Trstlist</i>
	<i>Settrust</i>
by object	<i>Trstlist</i>
	<i>Settrust</i>
for all volumes on a server	<i>Trstlist</i>
for selected volume objects	<i>Trstlist</i>
Restoring from file created by <i>trstlist</i>	<i>Settrust</i>

Setting

*Settrust*

## Managing objects (bindery)

### Operation

### Program

#### Groups

Creating

*Creatobj*

Deleting

*Grpadd**Delobj**Grpdel*

Determining membership in batch files

*Grpmemb*

Listing groups belonged to by user

*Usergrps*

Listing groups on a server

*Listobj*

Listing trustee assignments

for each group member

*Trstlist*

for the group object

*Trstlist*

#### Managers

adding managers

*Mgradd*

listing a group's managers

*Usermgrs*

listing managers

*Mgrlist*

removing managers

*Mgrdel*

#### Members

adding

*Grpadd*

deleting

*Makememb**Grpdel**Makememb*

listing

*Grplist*

#### Object ID

displaying

*Listobj*

#### Renaming

##### Full name

displaying

*Renobj*

setting

*Listobj**SetName*

Print queues - see printing section

Print servers - see printing section

#### Other object classes

Clearing connections

*Killconn*

Creating

*Creatobj*

Deleting

*Delobj*

Deleting a property

*Delprop*

Listing

all objects of any class

*Listobj*

by class

*Listobj*

object IDs

*Listobj*

---

	properties	<i>Scanprop</i>
	properties plus contents	<i>Scanprop</i>
Renaming		<i>Renobj</i>
Setting passwords		<i>Jrbpass</i>
		<i>Setpword</i>
Trustee assignments		<i>Trstlist</i>
Users		
	Account restrictions	
	checking restriction status in batch files via error levels	<i>Getrest</i>
	displaying	
	account balance	<i>Getbal</i>
	account disabled	<i>Getrest</i>
	account expired	<i>Getrest</i>
	account holds	<i>Getbal</i>
	account lock status	<i>Getrest</i>
	bad login address (intruder detection)	<i>Getrest</i>
	bad login count (intruder detection)	<i>Getrest</i>
	credit limit	<i>Getbal</i>
	force periodic password changes	<i>Getrest</i>
	expiration date	<i>Getrest</i>
	expired	<i>Getrest</i>
	grace logins allowed	<i>Getrest</i>
	grace logins remaining	<i>Getrest</i>
	last login date and time	<i>Getrest</i>
	maximum concurrent connections	<i>Getrest</i>
	minimum password length	<i>Getrest</i>
	next reset time (intruder detection)	<i>Getrest</i>
	password expiration date	<i>Getrest</i>
	password expiration interval	<i>Getrest</i>
	password required	<i>Getrest</i>
	previous login date and time	<i>Getrest</i>
	time restrictions	<i>Getrest</i>
	station restrictions	<i>Getrest</i>
	unique passwords required	<i>Getrest</i>
	user can change password	<i>Getrest</i>
	setting	
	account balance	<i>Setbal</i>
	account disabled	<i>Setrest</i>
	account expired	<i>Setrest</i>
	account lock status	<i>Setrest</i>
	bad login address (intruder detection)	<i>Setrest</i>
	bad login count (intruder detection)	<i>Setrest</i>
	credit limit	<i>Setbal</i>
	force periodic password changes	<i>Setrest</i>
	expiration date	<i>Setrest</i>
	expired	<i>Setrest</i>
	grace logins allowed	<i>Setrest</i>
	grace logins remaining	<i>Setrest</i>
	last login date and time	<i>Setrest</i>

---

	maximum concurrent connections	<i>Setrest</i>
	minimum password length	<i>Setrest</i>
	next reset time (intruder detection)	<i>Setrest</i>
	password expiration date	<i>Setrest</i>
	password expiration interval	<i>Setrest</i>
	password required	<i>Setrest</i>
	time restrictions	<i>Setrest</i>
	station restrictions	<i>Setrest</i>
	unique passwords required	<i>Setrest</i>
	user can change password	<i>Setrest</i>
Clearing connections		<i>Killconn</i>
Copying to another server		<i>Copyuser</i>
Creating users		<i>Creatobj</i>
		<i>Copyuser</i>
Deleting users		<i>Delobj</i>
Full name		
	converting to lowercase	<i>Setname</i>
	deleting	<i>Setname</i>
	displaying	<i>Listobj</i>
	setting	<i>Setname</i>
Group membership		
	adding	<i>Grpadd</i>
		<i>Makememb</i>
	removing from	<i>Grpdel</i>
		<i>Makememb</i>
	determining in a batch file	<i>Grpmemb</i>
	listing groups belonged to	<i>Usergrps</i>
Home directories		
	checking existence of	<i>Chkhome</i>
	checking rights to	<i>Chkhome</i>
	creating	<i>Creatobj</i>
		<i>Sethome2</i>
	deleting	<i>Sethome2</i>
	directory quotas on	
	displaying	<i>Dquota</i>
		<i>Getquota</i>
	removing	<i>Setquota</i>
	setting	<i>Setquota</i>
	HOME_DIR bindery property	
	checking the contents	<i>Chkhome</i>
	deleting	<i>Sethome2</i>
	displaying the contents	<i>Chkhome</i>
	setting	<i>Sethome2</i>
	mapping to the home directory	<i>Home2</i>
	space available in	<i>Dquota</i>
		<i>Getquota</i>
	space used by contents	<i>Dquota</i>
		<i>Getquota</i>



---

Intruder detection	
account locked status	<i>Getrest</i>
	<i>Setrest</i>
bad login address	<i>Getrest</i>
	<i>Setrest</i>
bad login count	<i>Getrest</i>
	<i>Setrest</i>
next reset time	<i>Getrest</i>
	<i>Setrest</i>
Last login time	<i>Getrest</i>
	<i>Lastlgn</i>
Listing	<i>Listobj</i>
Logged in	<i>Users</i>
Login scripts	
creating	
for existing users	<i>Lscripts</i>
when a user is created	<i>Creatobj</i>
displaying	<i>Lscripts</i>
displaying information in	
account balance	<i>Getbal</i>
account expiration date	<i>Accexp</i>
last login date and time	<i>Lastlgn</i>
quotas	<i>Dquota</i>
password expiration date	<i>Pwdexp</i>
space available	<i>Dquota</i>
space used	<i>Dquota</i>
providing warnings	
impending account expiration	<i>Accexp</i>
impending password expiration	<i>Pwdexp</i>
shortage of disk space	<i>Dquota</i>
viewing	<i>Lscripts</i>
Mail directories	
creating	<i>Creatobj</i>
	<i>Lscripts</i>
directory quotas on	
displaying	<i>Dquota</i>
	<i>Getquota</i>
removing	<i>Setquota</i>
setting	<i>Setquota</i>
space available in	<i>Dquota</i>
	<i>Getquota</i>
space used by contents	<i>Dquota</i>
	<i>Getquota</i>
Managers	
adding managers	<i>Mgradd</i>
listing a user's managers	<i>Usermgrs</i>
listing managers	<i>Mgrlist</i>
listing objects managed by a user	<i>Mgrlist</i>
removing managers	<i>Mgrdel</i>

---

Object ID	
displaying	<i>Listobj</i>
Passwords	
setting	
for a single user in a GUI	<i>Jrbpass</i>
across multiple servers	<i>Setpword</i>
to specified string	<i>Setpword</i>
to randomly generated string	<i>Setpword</i>
to user name	<i>Setpword</i>
when user is created	<i>Creatobj</i>
removing	<i>Setpword</i>
Print Job Configurations	
displaying	<i>Lscripts</i>
copying into mail directories	<i>Edit_pjc</i>
editing	<i>Lscripts</i>
	<i>Edit_pjc</i>
Renaming	
the user	<i>Renobj</i>
the home directory	<i>Renobj</i>
updating the HOME_DIR property	<i>Renobj</i>
Security equivalences	
checking for more than 32	<i>Grpadd</i>
displaying	<i>Makememb</i>
finding who is equivalent to an object	<i>Getequiv</i>
modified by SUPER.EXE	<i>Getequiv</i>
resetting modifications by SUPER.EXE	<i>Getequiv</i>
setting	<i>Setequiv</i>
Trustee assignments	<i>Setequiv</i>
	<i>Trstlist</i>
Work group managers	
Assigning work group manager status	<i>Mgradd</i>
Listing	
a manager's managed users	<i>Mgrlist</i>
work group managers	<i>Mgrlist</i>
Removing work group manager status	<i>Mgrdel</i>

## Managing objects (NDS)

### Operation

### Program

#### Aliases

##### Creating

*Creatobj*  
*Jrbimprt*

---

	<i>Move_obj</i>
	<i>Renobj</i>
Deleting	<i>Delobj</i>
Listing	<i>Listobj</i>
Application objects (ZEN)	
Adding associated objects	<i>Appadd</i>
Listing users of	<i>Applist</i>
Removing associated objects	<i>Appdel</i>
Directory maps	
Creating	<i>Creatobj</i>
Deleting	<i>Delobj</i>
Listing	<i>Listobj</i>
Domain groups (NDS for NT)	
Adding members	<i>Domgrps</i>
Creating local or global groups	<i>Domgrps</i>
Deleting local or global groups	<i>Domgrps</i>
Deleting members	<i>Domgrps</i>
Listing members	<i>Domgrps</i>
Domains (NDS for NT)	
Adding users	<i>Creatobj</i>
	<i>Domusers</i>
	<i>Jrbimprt</i>
Deleting users	<i>Domusers</i>
Listing users	<i>Domusers</i>
Groups	
Creating	<i>Creatobj</i>
	<i>Grpadd</i>
	<i>Jrbimprt</i>
Deleting	<i>Delobj</i>
	<i>Grpdel</i>
Determining membership in batch files	<i>Grpmemb</i>
Full name	
deleting	<i>Setname</i>
displaying	<i>Listobj</i>
setting	<i>Setname</i>
Listing groups belonged to by user	<i>Usergrps</i>
Listing groups in a container or NDS tree	<i>Listobj</i>
Members	
adding	<i>Grpadd</i>
	<i>Makememb</i>
deleting	<i>Grpdel</i>
	<i>Makememb</i>
listing	<i>Grplist</i>
Object ID	
displaying	<i>Listobj</i>

---

Renaming	<i>Getname</i>
Trustee assignments	<i>Renobj</i>
for all group members	<i>Trstlist</i>
for the group object	<i>Trstlist</i>
Other object classes not listed here	
ACLs	
displaying	<i>Setacl</i>
restoring	<i>Setacl</i>
setting	<i>Setacl</i>
Changing the case of the name	<i>Renobj</i>
Clearing connections	<i>Killconn</i>
Copying	<i>Copy_obj</i>
Creating	<i>Creatobj</i>
Creation date and time	<i>Getname</i>
Creator (eDir 8.5 onwards)	<i>Getname</i>
Deleting	<i>Delobj</i>
Deleting attributes	<i>Delattr</i>
Displaying all attribute values	<i>Getname</i>
Displaying rights to files and directories	<i>Fsrights</i>
Displaying rights to NDS objects	<i>Ndsrights</i>
Displaying selected attribute values	<i>Getname</i>
Listing	
all objects of any class	<i>Listobj</i>
by class	<i>Listobj</i>
object IDs	<i>Listobj</i>
Locating the NDS context	<i>Setcx</i>
Modification date and time	<i>Getname</i>
Modifier (eDir 8.5 onwards)	<i>Getname</i>
Moving within NDS	<i>Move_obj</i>
Renaming	<i>Renobj</i>
Security equivalences	
displaying	<i>Getequiv</i>
setting	<i>Setequiv</i>
Setting	
attributes holding object names	<i>Setname</i>
attributes of type "Back Link"	<i>Setname</i>
attributes of type "Boolean"	<i>Setname</i>
attributes of type "Counter"	<i>Setname</i>
attributes of type "Integer"	<i>Setname</i>
attributes of type "Interval"	<i>Setname</i>
attributes of type "Network Address"	<i>Setname</i>
	<i>Setrest</i>
attributes of type "Octet String"	<i>Setname</i>
attributes of type "Path"	<i>Setname</i>
attributes of type "Typed Name"	<i>Setname</i>
passwords	<i>Setpword</i>
proxy password attributes	<i>Setrest</i>
text attribute values	<i>Setname</i>

---

Trustee assignments	<i>Trstlist</i>
Organizational roles	
Adding occupants	<i>Orgroles</i>
Creating	<i>Creatobj</i>
Deleting	<i>Delobj</i>
Listing	<i>Listobj</i>
Listing occupants	<i>Orgroles</i>
Removing occupants	<i>Orgroles</i>
Printers (non NDPS)	
Adding queues	<i>Pradd</i>
Creating	<i>Creatobj</i>
	<i>Pradd</i>
Deleting	<i>Delobj</i>
Listing	<i>Listobj</i>
Removing queues	<i>Prdel</i>
Print queues	
Creating	<i>Creatobj</i>
	<i>Pqadd</i>
Deleting	<i>Delobj</i>
	<i>Pqdel</i>
Displaying disk usage in queue directories	<i>Getquota</i>
Listing	<i>Listobj</i>
Moving queue directories	<i>Pqmovdir</i>
Operators	
adding	<i>Pqadd</i>
listing	<i>Pqusers</i>
removing	<i>Pqdel</i>
Servicing print servers	
adding	<i>Pqadd</i>
listing	<i>Pqusers</i>
removing	<i>Pqdel</i>
Status	<i>Pqstatus</i>
Users	
adding	<i>Pqadd</i>
listing	<i>Pqusers</i>
removing	<i>Pqdel</i>
Print servers	
Account balances	
listing	<i>Getbal</i>
setting	<i>Setbal</i>
Associated printers	<i>Psusers</i>
Associated queues	<i>Psusers</i>
Creating	<i>Creatobj</i>
	<i>Psadd</i>
Credit limits	
listing	<i>Getbal</i>

---

	setting	<i>Setbal</i>
Deleting		<i>Delobj</i>
Listing		<i>Listobj</i>
Operators		
	adding	<i>Psadd</i>
	listing	<i>Psusers</i>
	removing	<i>Psdel</i>
Printers		
	adding	<i>Psadd</i>
	listing	<i>Psusers</i>
	removing	<i>Psdel</i>
Users		
	adding	<i>Psadd</i>
	listing	<i>Psusers</i>
	removing	<i>Psdel</i>
Profiles		
	Assigning users to use	<i>Profiles</i>
	Creating	<i>Creatobj</i>
		<i>Profiles</i>
	Deleting	<i>Delobj</i>
	Listing profile objects	<i>Listobj</i>
	Listing users	<i>Profiles</i>
	Login scripts	
	setting	<i>Lscripts</i>
	viewing	<i>Lscripts</i>
	Removing users from using	<i>Profiles</i>
Users		
	Account restrictions	
	displaying	
	account balance	<i>Getbal</i>
	account disabled	<i>Getrest</i>
	account expired	<i>Getrest</i>
	account holds	<i>Getbal</i>
	account lock status	<i>Getrest</i>
	bad login address (intruder detection)	<i>Getrest</i>
	bad login count (intruder detection)	<i>Getrest</i>
	credit limit	<i>Getbal</i>
	expiration date and time	<i>Getrest</i>
	expired	<i>Getrest</i>
	force periodic password changes	<i>Getrest</i>
	grace logins allowed	<i>Getrest</i>
	grace logins remaining	<i>Getrest</i>
	last login date and time	<i>Getrest</i>
	maximum concurrent connections	<i>Getrest</i>
	minimum password length	<i>Getrest</i>
	network addresses	<i>Getrest</i>
	next reset date and time (intruder detection)	<i>Getrest</i>
	password expiration date and time	<i>Getrest</i>

---

	password expiration interval	<i>Getrest</i>
	password required	<i>Getrest</i>
	previous login date and time	<i>Getrest</i>
	proxy password attributes	<i>Getrest</i>
	time restrictions	<i>Getrest</i>
	station restrictions	<i>Getrest</i>
	unique passwords required	<i>Getrest</i>
	user can change password	<i>Getrest</i>
setting		
	account balance	<i>Setbal</i>
	account disabled	<i>Setrest</i>
	account expired	<i>Setrest</i>
	bad login address (intruder detection)	<i>Setrest</i>
	bad login count (intruder detection)	<i>Setrest</i>
	credit limit	<i>Setbal</i>
	expiration date and time	<i>Setrest</i>
	expired	<i>Setrest</i>
	force periodic password changes	<i>Setrest</i>
	grace logins allowed	<i>Setrest</i>
	grace logins remaining	<i>Setrest</i>
	last login date and time	<i>Setrest</i>
	maximum concurrent connections	<i>Setrest</i>
	minimum password length	<i>Setrest</i>
	next reset date and time (intruder detection)	<i>Setrest</i>
	password expiration date and time	<i>Setrest</i>
	password expiration interval	<i>Setrest</i>
	password required	<i>Setrest</i>
	proxy password attributes	<i>Setrest</i>
	time restrictions	<i>Setrest</i>
	station restrictions	<i>Setrest</i>
	unique passwords required	<i>Setrest</i>
	user can change password	<i>Setrest</i>
ACLs		
	displaying	<i>Setacl</i>
	restoring	<i>Setacl</i>
	setting	<i>Setacl</i>
AD domains (via Novell Account Management)		
	adding	<i>Nadusers</i>
	listing	<i>Nadusers</i>
	removing	<i>Nadusers</i>
	setting passwords	<i>Setpword</i>
Attributes – displaying		<i>Getname</i>
Attributes – modification times		<i>Getname</i>
Auxiliary classes (NDS 8 onwards)		
	adding classes	<i>Jrbimprt</i>
		<i>Setname</i>
	setting associated attributes	<i>Jrbimprt</i>
		<i>Setname</i>

---

Clearing connections	
by connection number	<i>Killconn</i>
by user name	<i>Killconn</i>
holding a particular file open	<i>Openfile</i>
Copying	<i>Copy_obj</i>
Creating	<i>Creatobj</i>
	<i>Jrbimprt</i>
Creation dates and times	<i>Getname</i>
	<i>Getrest</i>
Creator (eDir 8.5 onwards)	<i>Getname</i>
	<i>Getrest</i>
Current context	
displaying	<i>Jrbcx</i>
setting	<i>Jrbcx</i>
	<i>Setcx</i>
Default server	
deleting	<i>Sethome2</i>
listing	<i>Chkhome</i>
setting	<i>Jrbimprt</i>
	<i>Sethome2</i>
Deleting	<i>Delobj</i>
	<i>Jrbimprt</i>
	<i>Delattr</i>
Deleting attributes	
Description	
deleting	<i>Setname</i>
displaying	<i>Getname</i>
setting	<i>Jrbimprt</i>
	<i>Setname</i>
Displaying all attributes	<i>Getname</i>
Displaying rights to files and directories	<i>Fsrightr</i>
Displaying rights to NDS objects	<i>Ndsrightr</i>
Full name	
converting to lowercase	<i>Setname</i>
deleting	<i>Setname</i>
displaying	<i>Getname</i>
	<i>Listobj</i>
setting	<i>Jrbimprt</i>
	<i>Setname</i>
Generational qualifier	
deleting	<i>Setname</i>
displaying	<i>Getname</i>
setting	<i>Jrbimprt</i>
	<i>Setname</i>
Given name	
deleting	<i>Setname</i>
displaying	<i>Getname</i>
setting	<i>Jrbimprt</i>
	<i>Setname</i>
Group membership	
adding	<i>Grpadd</i>



---

		<i>Jrbimprt</i>
		<i>Makememb</i>
	removing from	<i>Grpdel</i>
		<i>Makememb</i>
	determining in a batch file	<i>Grpmemb</i>
	listing groups belonged to	<i>Usergrps</i>
Home directories		
	checking the existence of	<i>Chkhome</i>
	checking trustee assignments	<i>Chkhome</i>
	creating	<i>Creatobj</i>
		<i>Jrbimprt</i>
		<i>Sethome2</i>
	creating subdirectories in	<i>Jrbimprt</i>
	deleting	<i>Delobj</i>
	directory quotas on	
	displaying	<i>Dquota</i>
		<i>Getquota</i>
	removing	<i>Jrbimprt</i>
		<i>Setquota</i>
	setting	<i>Jrbimprt</i>
		<i>Setquota</i>
	modifying trustee assignments	<i>Sethome2</i>
	renaming	<i>Renobj</i>
	setting directory attributes	<i>Jrbimprt</i>
	space available in	<i>Dquota</i>
		<i>Getquota</i>
	space used by contents	<i>Dquota</i>
		<i>Getquota</i>
Home directory attribute		
	checking the directory exists	<i>Chkhome</i>
	deleting	<i>Sethome2</i>
	displaying	<i>Chkhome</i>
	saving	<i>Chkhome</i>
		<i>Getname</i>
	restoring	<i>Sethome2</i>
	setting	<i>Jrbimprt</i>
		<i>Sethome2</i>
Initials		
	deleting	<i>Setname</i>
	displaying	<i>Getname</i>
	setting	<i>Jrbimprt</i>
		<i>Setname</i>
Internet email address		
	deleting	<i>Setname</i>
	displaying	<i>Getname</i>
	setting	<i>Jrbimprt</i>
		<i>Setname</i>

---

Intruder detection	
account locked status	<i>Getrest</i>
	<i>Setrest</i>
bad login address	<i>Getrest</i>
	<i>Setrest</i>
bad login count	<i>Getrest</i>
	<i>Setrest</i>
next reset date and time	<i>Getrest</i>
	<i>Setrest</i>
<i>Getrest</i>	
Last login time	<i>Getrest</i>
Language	
deleting	<i>Setname</i>
displaying	<i>Getname</i>
setting	<i>Jrbimprt</i>
	<i>Setname</i>
Listing	<i>Listobj</i>
Locating the NDS context	<i>Setcx</i>
Logged in	<i>Users</i>
Login scripts	
creating	
for existing users	<i>Lscripts</i>
when a user is created	<i>Creatobj</i>
	<i>Jrbimprt</i>
deleting	<i>Lscripts</i>
displaying	<i>Lscripts</i>
displaying information in	
account balance	<i>Getbal</i>
quotas	<i>Dquota</i>
space available	<i>Dquota</i>
space used	<i>Dquota</i>
providing warnings	
impending account expiration	<i>Accexp</i>
impending password expiration	<i>Pwdexp</i>
shortage of disk space	<i>Dquota</i>
Mail directories (SYS:MAIL)	
creating	<i>Creatobj</i>
	<i>Jrbimprt</i>
	<i>Lscripts</i>
directory quotas on	
displaying	<i>Dquota</i>
	<i>Getquota</i>
removing	<i>Setquota</i>
setting	<i>Setquota</i>
space available in	<i>Dquota</i>
	<i>Getquota</i>
space used by contents	<i>Dquota</i>
	<i>Getquota</i>
Modification dates and times	<i>Getname</i>
	<i>Getrest</i>
Modification of most attributes	<i>Jrbimprt</i>

---

Modifier (eDir 8.5 onwards)	<i>Getname</i>
Moving to another context	<i>Getrest</i>
Network address	<i>Move_obj</i>
deleting	<i>Deladdr</i>
displaying	<i>Delattr</i>
	<i>Getname</i>
	<i>Getrest</i>
NT domains (via Novell Account Management)	
adding	<i>Creatobj</i>
	<i>Domusers</i>
listing	<i>Jrbimprt</i>
removing	<i>Domusers</i>
setting passwords	<i>Domusers</i>
	<i>Jrbimprt</i>
	<i>Setpassword</i>
Object ID	
displaying	<i>Listobj</i>
	<i>Getname</i>
Passwords	
setting	
AD domain	<i>Setpassword</i>
in a GUI program	<i>Jrbpass</i>
NDS for NT	<i>Jrbimprt</i>
	<i>Jrbpass</i>
	<i>Setpassword</i>
NT domain	<i>Jrbimprt</i>
	<i>Jrbpass</i>
	<i>Pwdexp</i>
	<i>Setpassword</i>
simple passwords	<i>Jrbimprt</i>
	<i>Jrbpass</i>
	<i>Pwdexp</i>
	<i>Setpassword</i>
to specified string	<i>Setpassword</i>
to randomly generated string	<i>Setpassword</i>
to user name	<i>Setpassword</i>
when user is created	<i>Creatobj</i>
	<i>Jrbimprt</i>
workstation	<i>Jrbpass</i>
	<i>Pwdexp</i>
removing	<i>Setpassword</i>
verifying	<i>Setpassword</i>
Postal code	
deleting	<i>Setname</i>
displaying	<i>Getname</i>
setting	<i>Jrbimprt</i>
	<i>Setname</i>
Postal office box	
deleting	<i>Setname</i>

---

	displaying	<i>Getname</i>
	setting	<i>Jrbimprt</i>
		<i>Setname</i>
Profiles		
	adding	<i>Profiles</i>
	changing	<i>Profiles</i>
	removing	<i>Profiles</i>
Print job configurations		
	displaying	<i>Lscripts</i>
		<i>Edit_pjc</i>
	editing	<i>Edit_pjc</i>
Renaming		
	the user	<i>Renobj</i>
	the home directory	<i>Renobj</i>
	updating the "Home directory" attribute	<i>Renobj</i>
Security equivalences		
	displaying	<i>Getequiv</i>
	finding who is equivalent to an object	<i>Getequiv</i>
	setting	<i>Setequiv</i>
Surname		
	deleting	<i>Setname</i>
	displaying	<i>Getname</i>
	setting	<i>Jrbimprt</i>
		<i>Setname</i>
Title		
	deleting	<i>Setname</i>
	displaying	<i>Getname</i>
	setting	<i>Jrbimprt</i>
		<i>Setname</i>
Trustee assignments		<i>Trstlist</i>
		<i>Sethome2</i>
Workstation groups		
	Adding members	<i>Wgrpadd</i>
	Checking membership in batch	<i>Wgrpmemb</i>
	Creating	<i>Wgrpadd</i>
	Deleting members	<i>Wgrpdel</i>
	Listing	<i>Listobj</i>
	Listing members	<i>Wgrplist</i>
Workstation objects		
	Adding to workstation groups	<i>Wgrpadd</i>
	Deleting from workstation groups	<i>Wgrpdel</i>
	Listing logged in objects	<i>Users</i>
	Membership of workstation groups	<i>Wgrplist</i>

## Printing (NDPS)

### Operation

#### Printer Agents

Enqueued jobs  
     deleting  
     listing

*Pndeljob*  
*Pnlist*

Installed on a workstation  
     adding  
     deleting  
     listing

*Pnsetup*  
*Pnsetup*  
*Pnsetup*

#### Listing

anywhere in the tree  
 in a specified context  
  
 installed on a workstation

*Listobj*  
*Listobj*  
*Pnsetup*  
*Pnsetup*

#### Managers

adding  
 deleting  
 listing

*Pnadd*  
*Pndel*  
*Pnusers*

#### Operators

adding  
 deleting  
 listing

*Pnadd*  
*Pndel*  
*Pnusers*

#### Users

adding  
 deleting  
 listing

*Pnadd*  
*Pndel*  
*Pnusers*

## Printing (Queue based)

### Operation

#### Printers (NDS only)

Associated queues  
     adding  
     deleting

### Program

*Pradd*  
*Prdel*

---

Associated with print servers	
adding	<i>Psadd</i>
deleting	<i>Psdel</i>
listing	<i>Psusers</i>
Creating	<i>Creatobj</i>
	<i>Pradd</i>
Deleting	<i>Delobj</i>
Print queues	
Copying	<i>Pqcopy</i>
	<i>Copy_obj</i>
Creating	<i>Creatobj</i>
	<i>Pqadd</i>
Deleting	<i>Delobj</i>
	<i>Pqdel</i>
Jobs	
copying to another queue	<i>Pqcopy</i>
	<i>Pqmovjob</i>
deleting	
all jobs in a queue	<i>Pqdeljob</i>
by job number	<i>Pqdeljob</i>
by owner	<i>Pqdeljob</i>
listing	
all jobs in one or more queues	<i>Pqlist</i>
by user name	<i>Pqlist</i>
in queues across multiple servers	<i>Pqlist</i>
total jobs in a queue	<i>Pqlist</i>
moving to another queue	<i>Pqmovjob</i>
setting/releasing jobs on operator hold	<i>Pqreljob</i>
setting/releasing jobs on user hold	<i>Pqreljob</i>
Listing	<i>Listobj</i>
Moving the queue directory	<i>Pqmovdir</i>
Operators	
adding	<i>Pqadd</i>
deleting	<i>Pqdel</i>
listing	<i>Pqusers</i>
Renaming	<i>Renobj</i>
Servers	
adding	<i>Pqadd</i>
deleting	<i>Pqdel</i>
listing	<i>Pqusers</i>
Status	
attached print servers	<i>Pqstatus</i>
jobs can be submitted	<i>Pqstatus</i>
number of jobs in queue	<i>Pqstatus</i>
	<i>Pqlist</i>
servers can attach	<i>Pqstatus</i>
servers can service jobs	<i>Pqstatus</i>
Users	
adding	<i>Pqadd</i>

---

	deleting	<i>Pqdel</i>
	listing	<i>Pqusers</i>
Print servers		
	Clearing connections	<i>Killconn</i>
	Copying	<i>Pscopy</i>
		<i>Copy_obj</i>
	Creating	<i>Creatobj</i>
	Deleting	<i>Delobj</i>
	Listing	
	print servers logged in	<i>Users</i>
	print servers on a file server	<i>Listobj</i>
Operators		
	adding	<i>Psadd</i>
	deleting	<i>Psdel</i>
	listing	<i>Psusers</i>
Printers		
	adding (bindery)	<i>Psadd</i>
	adding (NDS only)	<i>Pradd</i>
	associated with the print server	<i>Psusers</i>
	deleting (NDS only)	<i>Prdel</i>
Queues		<i>Psusers</i>
Renaming		<i>Renobj</i>
Setting description (full name)		<i>Setname</i>
Users		
	adding	<i>Psadd</i>
	deleting	<i>Psdel</i>
	listing	<i>Psusers</i>

## Server Operations

### Operation

### Program

Attaching to servers	<i>Mattach</i>
Bindery (NW 3.x)	
Opening	<i>Bindopen</i>
Closing	<i>Bindclos</i>
Connections	
Clearing	
by connection number	<i>Killconn</i>
by connection status	<i>Killconn</i>
by login period	<i>Killconn</i>
by object name	<i>Killconn</i>
by object class	<i>Killconn</i>
holding a particular file open	<i>Openfile</i>
logged-in connections	<i>Killconn</i>
not logged-in connections	<i>Killconn</i>

---

Listing	
all connections	<i>Users</i>
by object name	<i>Users</i>
by object class	<i>Users</i>
connection type (bindery, licensed, unlicensed)	<i>Users</i>
connection type (IPX, TCP/IP, AFP)	<i>Users</i>
requests to server	<i>Users</i>
	<i>Requests</i>
Console Operators	
Adding	<i>Con_ops</i>
Deleting	<i>Con_ops</i>
Listing	<i>Con_ops</i>
Detaching from servers	<i>Mattach</i>
Displaying server information	
License counts	<i>Mattach</i>
Listing file servers on the network	<i>Listobj</i>
	<i>Mattach</i>
NetWare version	<i>Mattach</i>
Serial numbers	<i>Mattach</i>
Up time	<i>Mattach</i>
Downing the server	<i>Downsrvr</i>
	<i>Serv_cmd</i>
Intruder detection (NW 3.x)	
Displaying the server settings	<i>Getrest</i>
Modifying the server settings	<i>Setrest</i>
Issuing server commands (NW 4.x onwards)	
Adding a name space to a volume	<i>Serv_cmd</i>
Dismounting a volume	<i>Serv_cmd</i>
Displaying cache information	<i>Serv_cmd</i>
Displaying cpu information	<i>Serv_cmd</i>
Displaying directory cache information	<i>Serv_cmd</i>
Displaying general information	<i>Serv_cmd</i>
Displaying lsl information	<i>Serv_cmd</i>
Displaying loaded modules	<i>Serv_cmd</i>
Displaying known networks	<i>Serv_cmd</i>
Displaying known servers	<i>Serv_cmd</i>
Displaying memory information	<i>Serv_cmd</i>
Displaying NCP information	<i>Serv_cmd</i>
Displaying SET parameter values	<i>Serv_cmd</i>
Displaying login and TTS status	<i>Serv_cmd</i>
Executing an NCF file	<i>Serv_cmd</i>
Issuing SET commands	<i>Serv_cmd</i>
Loading an NLM	<i>Serv_cmd</i>
Mounting a volume	<i>Serv_cmd</i>
Unloading an NLM	<i>Serv_cmd</i>
Managing bindery context settings	<i>Serv_cmd</i>
Logins	
Disabling	<i>Disable</i>
	<i>Serv_cmd</i>
Enabling	<i>Enable</i>



---

NDS schema	<i>Serv_cmd</i>
Displaying attributes	<i>Schema</i>
Displaying object classes	<i>Schema</i>
Volumes	
Displaying information	
creation date and time	<i>Vol_info</i>
name spaces loaded	<i>Vol_info</i>
size, space used and free	<i>Vol_info</i>
blocks occupied by deleted files	<i>Vol_info</i>
directory entries	<i>Vol_info</i>
sub allocation and compression info	<i>Vol_info</i>
type (traditional or NSS)	<i>Vol_info</i>
volumes on each server	<i>Vol_info</i>
Setting the creation date	<i>Setvdate</i>

## Workstation Operations

<u>Operation</u>	<u>Program</u>
Connections to servers	
Attaching	<i>Mattach</i>
Detaching	<i>Mattach</i>
Listing connections	<i>Mattach</i>
Current context	
Displaying	<i>Jrbcx</i>
Setting	<i>Jrbcx</i>
	<i>Setcx</i>
Current tree	
Displaying	<i>Jrbcx</i>
Setting	<i>Jrbcx</i>
Drive mappings	
Deleting a drive mapping	<i>Jrbmap</i>
Displaying	<i>Jrbmap</i>
Mapping a network drive	<i>Jrbmap</i>
Mapping a local drive to a network drive	<i>Jrbmap</i>
Mapping to a queue's directory	<i>Home2</i>
Mapping to a print server's directory	<i>Home2</i>
Mapping to a user's home directory	<i>Home2</i>
Mapping to a user's mail directory	<i>Home2</i>
Saving and restoring	<i>Jrbmap</i>
Identifying the DOS client type	
Client32	<i>CL_type</i>
NETX	<i>CL_type</i>
	<i>Is_vlms</i>
VLMS	<i>CL_type</i>
	<i>Is_vlms</i>

## NDPS printers

- adding installed
- deleting installed
- listing installed
- setting the default

*Pnsetup*  
*Pnsetup*  
*Pnsetup*  
*Pnsetup*

## GroupWise

### Operation

Adding users

Distribution lists

- Adding members

- Changing the participation type

- Listing membership
- Removing users

Removing users

Renaming users

Setting mailbox expiration time

Setting passwords

Setting visibility

### Program

*Creatobj*  
*Jrbimprt*

*Gwdlists*  
*Jrbimprt*  
*Gwdlists*  
*Jrbimprt*  
*Gwdlists*  
*Gwdlists*  
*Jrbimprt*

*Delobj*  
*Jrbimprt*  
*Renobj*

*Jrbimprt*  
*Setrest*  
*Jrbimprt*  
*Setpword*  
*Jrbimprt*  
*Setrest*

## Acknowledgements

1. Dr Alan McKinnon, Director of the Centre for Computing and Biometrics at Lincoln University for assistance in obtaining ownership of the original utilities from Lincoln University.
2. David Harris for permission to include his SETHOME and HOME utilities, for permission to write SETHOME2 and for lots of sound advice.
3. Mark Cramer for his meticulous testing of JRButils. The product has benefited enormously from his attention to detail and suggestions for enhancements.
4. Craig Saunders and the team at DigitalFusion, Christchurch, NZ, for production of the setup programs and other projects.
5. Craig Benbow for his work on the fully graphical utilities in Part E.
6. Lots of people for their support and for providing invaluable feedback including bug reports and suggestions for new features and new programs. In particular, Brian Cassidy, Al Lowe-Norris, Herb McBride, Juergen Rosemeyer and Jaap van Ginkel.