

TME® 10 Software Distribution
for AIX®



Command Reference

Version 3.1.5

TME® 10 Software Distribution
for AIX®



Command Reference

Version 3.1.5

Contents

What's New in This Release	xi
Support for New Platforms	xi
New Pristine Scenarios	xi
Complete Platform Support Table	xi
Deletion of Pending Requests from Host	xiii
Changes to Statuses Reported by 'stattg'	xiv
 Chapter 1. Command-Line Operations	 1
Software Distribution Commands	1
Help Command	1
Configuration Commands	2
Building and Distribution Commands	4
Change Control Commands	5
System Administration Commands	6
Request Management Commands	7
Queue Management Commands	8
Transmission Plan Management Commands	9
Entering a User ID and Password	9
Entering Commands	9
Entering Data in the Command-Line Interface	11
Entering Target Names	12
Entering Global Names	13
Entering Dates and Times	13
Recursive Parameters	16
Running a Procedure after a Request Has Completed	16
Return Codes and Error Messages	17
Authorization Levels for Command Use	18
 Chapter 2. TME 10 Software Distribution Line Commands	 21
acc – Accept an Installed Change File	22
act – Activate Changes on a Target	27
addgp – Add a Target Group	31
addpln – Add a Plan	33
addpm – Add Parameters to a Target	34
addprf – Add User Profile	36
addtg – Create a Target	39
addusr – Create a User	45
auth – Authorize Targets to Access Files	47
bld – Build a Change File on the Server	51
cat – Create a Catalog Entry	52
connect – Establish Mobile Connection with Server	56
del – Delete a File	58
delcm – Delete a Change File Status Record	62
delf – Delete a Noncataloged File	63
delgp – Delete a Target Group	67

delpm – Delete Parameters from a Target	69
delpf – Delete an Authorization Profile	70
delrq – Delete a Request	71
deltg – Delete a Target	73
delusr – Delete a User	74
disconnect – Close Mobile Connection with Server	75
eraserq – Erase Requests	76
exec – Execute a Program or Script	78
execf – Execute a Noncataloged Program or Script	82
execpln – Execute a Plan	86
exp – Copy a File from the Server to Your Workstation	89
help – Display Help for the Command-Line Interface	90
hldc – Hold Remote Communication	91
hldq – Hold One or All Queues	92
hldrq – Hold a Request	94
hldtg – Hold a Target	96
imp – Import a Change File into the System	97
inst – Install a Change File	99
inv – Inventory Discovery	106
log – Display the Contents of the Message Log	111
ls – List Entries in the Catalog	114
lsak – List Access Keys	116
lsbs – List Configuration	117
lscf – List the Contents of a Change File	118
lscm – Report the Status of Files	121
lsgp – List Target Groups	124
lspln – List a Plan	126
lsprf – List Profiles	127
lsq – Display Queue Entry Details	129
lsrq – List Change Control and Distribution Requests	132
lstg – List Targets	137
lsusr – List User Authorizations	141
nvdn – Start a Command Line Session	143
open – Open a Connection to a TME 10 Software Distribution Server	144
preqpln – Generate a Plan for a Change File Prerequisites	145
prgq – Purge One or All Entries from a Queue	148
pqty – Give Priority to a Queue	150
quit – Close an Interactive Command Line Session	152
relc – Release Remote Communication	153
relq – Release One or All Queues	154
relrq – Release a Request	156
reltg – Release a Target	158
rem – Remove a Change File From a Target or Group	159
rentg – Rename a Target	164
reset – Reset TME 10 Software Distribution	166
rld – Reload Configuration Information	167
rstrq – Restart a Request	168
rtv – Retrieve a File	169

rtrvf – Retrieve a Noncataloged File	174
send – Send a File to Another Target	179
sendf – Send a Noncataloged File	184
start – Start TME 10 Software Distribution on Your Workstation	189
stat – Check the Status of a Communication Queue	190
stattg – List Target Status	193
stop – Stop TME 10 Software Distribution on Your Workstation	196
svr – Select a Server	197
trace – Set Internal Tracing	198
troff – Stop a Trace	200
tron – Start a Trace	201
unauth – Remove a Target Authorization to Access a File	202
unbld – Unpack the components of a change file	206
uncat – Remove a Catalog Entry	209
uninst – Remove a Component from a Target	210
upd – Update a Catalog Entry	215
updak – Update an Access Key	216
updb – Update Configuration	217
updbcm – Update Change Control Status	218
updpn – Update a Plan	223
updpf – Update a User Authorization Profile	224
updpwd – Update a User Password	227
updrq – Update a Request	228
updtg – Update Targets	232
updusr – Update a User Definition	239
vercm – Verify Change Control Status	241
 Chapter 3. Creating Change File Profiles	 247
Static and Dynamic Change File Profiles	247
The Structure of a Change File Profile	247
Entering Lines in a Profile	250
Rules for Entering Keywords	250
Change-File Profile Keywords	250
Envelope Section Keywords	250
Product Section Keywords	257
Remote Directory Section Keywords	258
Object Section Keywords	259
Dynamic Section Keywords	264
Examples of TME 10 Software Distribution Change File Profiles	265
Example of an AIX installp Change File Profile	267
Example of AIX installp Change File Profile Objects	268
Example of AIX installp -l Output	269
 Index	 271

Figures

1.	The structure of a Change File profile	248
2.	Example of a TME 10 Software Distribution Change File profile	266
3.	Example of a TME 10 Software Distribution Change File profile	267
4.	AIX installp Change File profile	268
5.	AIX installp Change File profile objects	268
6.	Example of AIX installp -l output	269

Tables

1.	TME 10 Software Distribution, Version 3.1.5 Platform Support	xii
2.	Products from which TME 10 Software Distribution, Version 3.1.5 can be upgraded	xiii
3.	SBCS and MBCS Supported Characters	11
4.	Authorization Levels for Command Use	18
5.	Default File System IDs	53
6.	Object Types and their SNA/FS Class Codes	53
7.	Object Types and their SNA/FS Class Codes	114
8.	Default File System IDs	185
9.	Values for Component Names	199
10.	Updating User Authorizations to Targets	240

What's New in This Release

Release 3.1.5 of TME 10 Software Distribution contains the following new or changed functionalities:

Support for New Platforms

TME 10 Software Distribution, Version 3.1.5 adds support for the following platforms:

- Windows 2000 (Professional and Server)
- Windows NT 4.0 (Service Pack 5 and 6a)
- OS/2, version 4.5 (Warp server for e-business)
- AIX, version 4.3.x

New Pristine Scenarios

TME 10 Software Distribution, Version 3.1.5 Client can be installed on a pristine workstation in the following environments:

- Windows 2000 Professional
- Windows 2000 Server
- Windows NT 4.0 Server/Workstation
- OS/2 4.5 (Warp Server for e-business)
- AIX 4.3.3

This is in addition to the following pristine installation environments, which are maintained from the previous release:

- Windows 3.11
- Windows 95
- Windows NT Version 3.51
- OS/2 3.0.x (Warp)

Complete Platform Support Table

Table 1 on page xii shows details of the platforms on which TME 10 Software Distribution is available. The columns in the table contain the following information:

Server Scratch	Indicates whether the Server software can be installed from scratch. Scenarios describing how to carry out the scratch installations can be found in the relevant Quick Beginnings manuals.
Server Upgrade	Indicates which version of the TME 10 Software Distribution Server can be upgraded, by supplying a reference that can be looked up in Table 2 on page xiii. Scenarios describing how to carry out the upgrade can be found in the README file.
Client Scratch	Indicates whether the Client software can be installed from scratch. Scenarios describing how to carry out the scratch installations can be found in the Client Installation and Customization manual.

What's New in This Release

Client Pristine	Indicates whether the Client software can be installed on a pristine workstation (i.e. a workstation with no operating system installed). Scenarios describing how to carry out the pristine installations can be found in the Pristine and Migration Scenarios manual or the Installation Scenarios for AIX manual.
Client Upgrade	Indicates which version of which Client software can be upgraded, by supplying a reference that can be looked up in Table 2 on page xiii. Scenarios describing how to carry out the upgrade can be found in the relevant README files.

<i>Table 1. TME 10 Software Distribution, Version 3.1.5 Platform Support</i>						
Platform		Server		Client		
OS	Version	Scratch	Upgrade	Scratch	Pristine	Upgrade
Windows	2000 Professional	Y		Y	Y	
	2000 Server	Y		Y	Y	
	NT 4.0 (SP5 & 6a)	Y	1	Y	Y	5
	NT 3.51	Y	1	Y	Y	5
	98			Y		6
	95			Y	Y	6
	3.11			Y	Y	7
OS/2	3.0x	Y	2	Y	Y	8, 11
	4.0	Y	2	Y		8, 11
	4.5 (Warp server for e-business)	Y		Y	Y	
AIX	3.2.5 - 4.2.1	Y	3	Y		9
	4.3.3	Y	3	Y	Y	9
NetWare	4.11 - 4.2x	Y	4	Y		10

Table 2 on page xiii shows the products (and versions) that can be upgraded to TME 10 Software Distribution, Version 3.1.5; the Reference column refers to Table 1.

Table 2. Products from which TME 10 Software Distribution, Version 3.1.5 can be upgraded

Reference (see Table 1)	Version installed	CSD or Fix Pack installed
TME 10 Software Distribution		
1	3.1.3 Server for Windows NT	XR21923
2	3.1.3 Server for OS/2	XR21923
3	3.1.4 Server for AIX	99/10
4	3.1.3 Server for NetWare	XR21924
5	3.1.3 Client for Windows NT	XR21923
6	3.1.3 Client for Windows 9x	XR21923
7	3.1.3 Client for Windows 3.1	XR21923
8	3.1.3 Client for OS/2	XR21923
9	3.1.4 Client for AIX	99/10
10	3.1.3 Client for NetWare	XR21924
NetView DM/2		
11	2.1	

Deletion of Pending Requests from Host

In the circumstances where TME 10 Software Distribution is executing software distribution requests from a focal point running Tivoli NetView Distribution Manager (NetView DM) Release 7, the MVS focal point can now issue a request to delete any distribution requests that are waiting to be processed or are being processed at the TME 10 Software Distribution server.

- In the case of a distribution request waiting to be processed, the original request will be deleted, and a report sent to the focal point confirming the deletion.
- In the case of a distribution request that is in execution when the deletion request arrives, the original request will be completed, and a report sent to the MVS focal point confirming the successful completion of the original request; no report concerning the unfulfilled deletion request will be sent.

In the case of nodes in a distribution network that are not running TME 10 Software Distribution, Version 3.1.5 (i.e. older versions of TME 10 Software Distribution or NetView DM/2) the deletion requests from the MVS focal point will be ignored.

This functionality runs in the background with no intervention required by the operator of the TME 10 Software Distribution server.

Note: As a consequence of this new functionality global names starting with `$DELETE.$PENDING` are reserved, and may not be used.

Changes to Statuses Reported by 'stattg'

The `stattg` command gives details of the status of the agent at the local target. A new parameter has been added to the command to reveal additional information.

In the previous releases, and when used without the new parameter, the command reports these statuses:

<i>Available</i>	Agent running and ready to process a request
<i>Not Available</i>	Agent not running or not accessible
<i>Busy</i>	Agent running a request and not available to process any other request.

There are circumstances in which it is possible for the server to have in its database more than one workstation name for the same agent.

For example, if a workstation has been re-defined to the server for some reason, the operator may have supplied a different workstation name than that originally used, but have used the original hostname. In this event, the agent now has the new workstation name, but the server has both workstation names defined; prior to this release the agent reported itself as being *Available* under both workstation names.

With this release, by using the parameter `-c`, in the event that the agent is *Available* and not *Busy*, the command now returns the status *Unknown* if the hostname of the agent is correct but the workstation name in the status request does not match the workstation name of the agent. Thus, by using the `-c` parameter, polling both workstation names will allow you to identify which is the correct one, as one will return the status *Available* and the other *Unknown*. If the parameter is not used, the original functionality is maintained.

However, before using this parameter you should consider the question of the timing of the `stattg` requests. When an agent receives a `stattg` request it sends the status to the server but is then not immediately available to satisfy another request. This means that a second request, received within, say, one minute of the first request, will return the status *Not Available*. If you are polling two suspect workstation names you should wait for this period before sending the second request.

This also means that if you send a `stattg` request using the asterisk wildcard to obtain the status of all or a group of workstations, the results received will depend on whether the *incorrect* workstation name comes before or after the *correct* one in the server's database:

Incorrect workstation name is polled first

The status of the *incorrect* workstation name will be given as *Unknown*, while the *correct* workstation will give *Not Available*

Correct workstation name is polled first

The status of the *correct* workstation name will be given as *Available* while the *incorrect* workstation will give *Not Available*

Thus, after using the asterisk wildcard with the `-c` parameter, you should individually poll each workstation name given as *Not Available*, waiting for approximately one minute before issuing each command. Workstations that are genuinely unavailable will report the same status as before; workstations that were unavailable while they were recovering from a previous `stattg` command will now report their true status.

The full details of the `stattg` command are given in *TME 10 Software Distribution Command Reference*, *TME 10 Software Distribution for NetWare Command Reference* and *TME 10 Software Distribution for AIX Reference*.

What's New in This Release

Chapter 1. Command-Line Operations

This chapter describes the purpose and syntax of the commands.

Several services of the product can be accessed from the command line. The following sections describe how these services can be accessed from a command line, as well as the various parameters associated with their use.

Software Distribution Commands

The software distribution command-line interface supports the commands listed in this section.

You can also find the following information:

- A description of the command syntax
- How to enter target names
- How to enter global names
- How to enter the date and time
- How to interpret return codes and error messages
- A summary of how user-group authorization affects the availability of commands

Help Command

To see a full list of the available commands, enter:

```
nvdm help
```

To see the full list of parameters for a particular command, specify the name of the command as well. For example, to find the parameters for the **rentg** command that renames a target, enter:

```
nvdm help rentg
```

In this environment, you can also display online the commands documented in this chapter. To display the information for a specific command, enter:

```
man nvdm_<cmd>
```

where <cmd> is the name of the command you want information for.

Configuration Commands

Use these commands to control the configuration of targets, groups, remote targets, and installation parameters in the local network.

addgp (see 31)

Define a group of targets

addpm (see 34)

Add parameters to a target definition

addprf (see 36)

Create a new authorization profile

addtg (see 39)

Add a new target workstation to the system

addusr (see 45)

Create a new TME 10 Software Distribution user

delgp (see 67)

Delete a group of targets

delpm (see 69)

Delete parameters from a target definition

delprf (see 70)

Erase a user authorization profile

deltg (see 73)

Remove a target definition

delusr (see 74)

Delete a user definition

inv (see 106)

Inventory discovery

lsak (see 116)

List names and descriptions of data or target access keys

lsbs (see 117)

List the server configuration

lsgp (see 124)

List groups of targets

lsprf (see 127)

List the contents of a user authorization profile

lstg (see 137)

List target configurations

lsusr (see 141)

List user authorization levels

rentg (see 164)

Rename a target workstation

updak (see 216)

Update a data or target access key

updcn (see 218)

Update the change management status for an object at a target

updbn (see 217)

Change the server configuration

updprf (see 224)

Update a user authorization profile

updpwd (see 227)

Update a user's password

updtg (see 232)

Update the target configuration

updusr (see 239)

Change user authorization levels

Command-Line Interface

Building and Distribution Commands

Use these commands to build change files and distribute data and change files.

bld (see 51)

Build a change file

cat (see 52)

Add an entry to the catalog

del (see 58)

Delete a file

delf (see 63)

Delete a noncataloged file

exp (see 89)

Export a change file

imp (see 97)

Import and catalog a change file

ls (see 114)

List entries in the catalog

lscf (see 118)

List the contents of a change file

rtrv (see 169)

Retrieve a file from a target

rtrvf (see 174)

Retrieve a noncataloged file from a target

send (see 179)

Send a file to a target

sendf (see 184)

Send a noncataloged file to a target

unbld (see 206)

Unpack the components of a change file

uncat (see 209)

Remove an entry from the catalog

upd (see 215)

Update a catalog entry

updcn (see 218)

Update the change management status for an object at a target

Change Control Commands

Use these commands for change control requests.

acc (see 22)

Accept a previously installed change file on a target

act (see 27)

Activate changes on a target

auth (see 47)

Authorize a change file for use by a target

delcm (see 62)

Clear the status of change files

exec (see 78)

Execute a command procedure on a specified target

execf (see 82)

Execute a noncataloged program or script on a target

inst (see 99)

Request the installation of a change file on a target

lscm (see 121)

List the status of change files at one or more targets

rem (see 159)

Remove a previously installed change file from a target

unauth (see 202)

Remove the authorization for a target to use a change file

uninst (see 210)

Completely remove a component from a target

vercm (see 241)

Verify the status of change files

Command-Line Interface

System Administration Commands

Use these commands to control product behavior.

connect

Connect to a server from a mobile client

disconnect

Halt the connection of a mobile client with a server

hldtg (see 96)

Hold a push-mode target

log (see 111)

View the message log

reset (see 166)

Reset the product

rld (see 167)

Prompt all components to reread the configuration

reltg (see 158)

Release a push-mode target

start (see 189)

Start the product

stat (see 190)

Report status of TME 10 Software Distribution communication

stattg (see 193)

List the status of targets

stop (see 196)

Stop the product

svr (see 197)

Select which server to operate on

trace (see 198)

Set internal tracing

troff (see 200)

Stop tracing

tron (see 201)

Start tracing

Request Management Commands

Use these commands to manage requests.

delrq (see 71)

Delete a submitted request

eraserq (see 76)

Erase a request from the database

hldrq (see 94)

Hold a request

lsrq (see 132)

List all requests on the database

relrq (see 156)

Release a request

rstrq (see 168)

Restart a request

updrq (see 228)

Update a request

Command-Line Interface

Queue Management Commands

Use these commands to manage the distribution queues on the server.

hldc (see 91)

Hold remote communication

hldq (see 92)

Hold processing distributions on a queue

lsq (see 129)

List the distributions waiting on a queue

prgq (see 148)

Remove a distribution from a queue

prtyq (see 150)

Increase the priority of an entry in a queue

relc (see 153)

Release remote communication

relq (see 154)

Release processing distributions on a queue

Transmission Plan Management Commands

Use these commands to handle transmission plans.

addpln	Create a plan
execpln	Execute a plan
lspln	List the contents of a plan
preqpln	Generate a plan for a change file prerequisites
updpln	Update a plan

Entering a User ID and Password

When you enter a command from the command-line interface, you must supply your user ID and password, if any. You can supply this information using one of the following methods:

- Enter your user ID and password as environment variables using the FNDUSER and FNDPASSWORD keywords. At an AIX command line, enter:


```
export FNDUSER=<userid>
export FNDPASSWORD=<password>
```

 where <userid> is your user ID and <password> is the associated password, if any. You can set the value of FNDUSER to * (asterisk), meaning that every time you enter a command you are prompted for your user name, and password if any. If you do not set any value for FNDUSER, it defaults to the user ID you used to log on to the operating system, (which may not be valid for TME 10 Software Distribution).
- Enter your user ID and password as parameters when you enter a command:

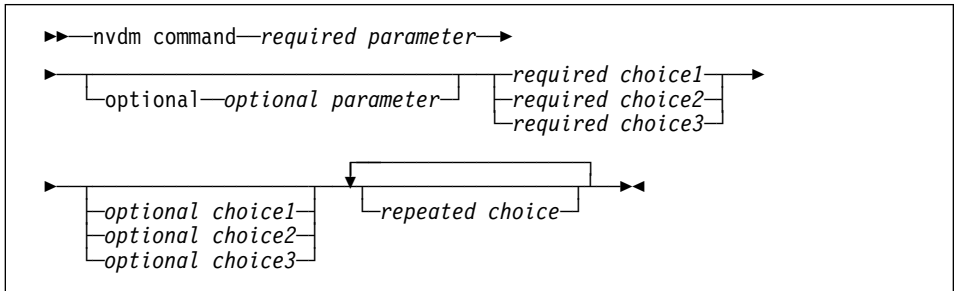

```
nvdm cmd_name <cmd_parameters> -u<user_name> -p<password>
```
- Open an *interactive command line session* by entering the **nvdm** command. When you issue this command, you are prompted to specify your user ID and password. You then no longer need to supply this information for each command you enter. The interactive command line session stays open until you close it by issuing the **quit** command. See “nvdm – Start a Command Line Session” on page 143 and “quit – Close an Interactive Command Line Session” on page 152.

Entering Commands

This section describes the syntax diagrams used to document commands. Before you attempt to consult or use any of the commands, it is important that you understand both the notation used to define them and the limitations that apply to the parameters.

Command-Line Interface

The command syntax describes the format in which the commands are entered into the command-line interface. They take the following form:



nvdm identifies the TME 10 Software Distribution product. It can be entered in upper or lower case, but mixed case is not supported.

command identifies the TME 10 Software Distribution command to be run. The commands are described in subsequent sections. They can be entered in uppercase, lowercase, or mixed case.

Parameters to a command are mandatory or optional. *Mandatory* parameters must be specified in the order shown in the description of the commands.

Optional parameters can be entered in any order, and in any position after the command, even before the mandatory parameters.

Optional parameters need not be entered. The description of the commands describes the action taken if each optional parameter is omitted. Optional parameters have the form:

—x option

—x is the parameter for which the value x changes to identify each different parameter; for example —w for a target workstation. Parameters may be upper- or lowercase. If an optional parameter has associated data, that data is mandatory if the parameter is present. Spaces are allowed between the parameter and the start of the option. For clarity, spaces are used to illustrate the syntax in this document.

If the data for any optional or mandatory parameters includes special characters (such as spaces or asterisks), you should enclose the data in delimiting characters to avoid it being misinterpreted by your command shell. The delimitation characters that you use depend upon your shell, but typically either quotation marks (") or apostrophes (') are used. If a command option is a string composed of two or more words, it must be included in quotation marks as well.

For more details about how to prevent your command shell from misinterpreting your commands, refer to the description of the shell in the manuals supplied with your computer.

Entering Data in the Command-Line Interface

Throughout this book, restrictions about the length of fields are made with respect to single byte character set (SBCS) languages. For double and multiple byte character set (DBCS and MBCS) languages, those restrictions are calculated in terms of bytes, not characters.

For example, a field that can have up to eight characters in an SBCS language is limited to eight bytes in DBCS and MBCS languages.

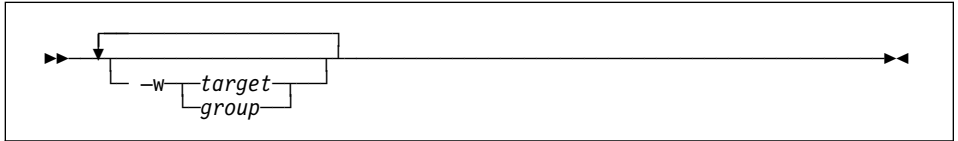
To avoid unpredictable results when exchanging data between workstations that use different character sets, it is suggested that you use alphabetic characters (lowercase and uppercase), digits, and the characters listed in Table 3, unless otherwise stated.

<i>Table 3. SBCS and MBCS Supported Characters</i>			
Symbol	Description	Symbol	Description
blank	space	!	exclamation mark
?	question mark	#	number sign
\$	dollar sign	%	percent
&	ampersand	'	apostrophe
(left parenthesis)	right parenthesis
*	asterisk	+	plus sign
,	comma	-	hyphen
.	period	/	slash
:	colon	;	semicolon
<	less than	=	equal sign
>	greater than	"	quotation mark
@	commercial at	[left bracket
\	backslash]	right bracket
^	circumflex	_	underscore
`	grave accent	{	left brace
	vertical line	}	right brace
~	tilde		

Command-Line Interface

Entering Target Names

Many commands apply to a specific target or to a group of targets. In these commands the targets are identified using the target workstation option (**-w**) or a parameter. This takes the form:



The following rules apply when you specify targets or groups of targets:

- If you omit the **-w** option, your target is used as the default.
- If you supply the **-w** option with a *target* name, the command applies only to that target. You can specify up to 16 target names.
A shorthand method for referring to your server is to specify **-w \$**.
- If you supply the **-w** option with *group* names, the command applies to all targets in the groups. You can specify up to 16 group names.
- Do not use the **-w** option in base systems for commands that apply to local targets.
- For change control commands, the **-w** option is valid only for users authorized to manage all targets. Other users must omit the option, meaning that they can issue such commands only for their own target.

With some of the commands you can use an asterisk (*) enclosed in single or double quotation marks to represent all targets. (See also “Entering Global Names” on page 13). These commands are:

- delcm
- delrq
- log
- lscm
- lsgp
- lsrq
- lstg
- stattg

You can use the following tokens for targets in plans:

\$(TARGETLIST)

This token is expanded to the list of target names passed as an argument of the `execpln` command.

\$(SERVERLIST)

This token is expanded to the list of server names passed as an argument of the `execpln` command.

\$(ORIGINATOR)

This token is expanded to the name of the target where the `execpln` command originated.

Entering Global Names

Many commands take a global name as a parameter. Global names consist of between two and ten tokens. In some cases, wildcard characters can be used in place of any token. The following wildcard characters are recognized:

*	Any string. Matches to end of name if used in the last token.
*.	Matches to end of that token only
?	Any character
(H)	Highest value
(L)	Lowest value

If wildcard characters are allowed, this is stated in the parameter description. Sometimes, when wildcard characters are allowed, they must identify a unique file. This is also stated in the parameter description. You may need to enclose `*` within delimiters (enter either `'*'`, `/*`, or `"**"`) to prevent your operating system version from misinterpreting them.

If a global name refers to a change file, it must have one of the following forms:

- `compid.REF.level.version`
- `compid.UPD.newlevel.oldlevel.version`
- `compid.FIX.level.fix.version`

In these examples:

- *compid* can be between 1 and 6 tokens in length
- *level*, *newlevel*, and *oldlevel* must be numeric
- *newlevel* must be greater than *oldlevel*

When you type a global name, you can enter the characters in either uppercase or lowercase. However, they are converted to uppercase when you press the Enter key.

Entering Dates and Times

To execute a request at a certain date and time, specify the following parameters:

Schedule date (**-d**) and time (**-t**)

-d *schedule date*

The date on which the request is to be processed and validated at the workstation from which you submit the request, the *origin* workstation.

-t *schedule time*

The time at which the request is to be processed and validated at the origin workstation.

Execution date (**-db**) and time (**-tb**)

Command-Line Interface

-db *request execution date*

The date on which the request is to be executed at the destination workstation.

-tb *request execution time*

The time at which the request is to be executed at the destination workstation.

Expiration date (**-da**) and time (**-ta**)

-da *request expiration date*

The date after which the request cannot be executed at the destination workstation.

-ta *request expiration time*

The time after which the request cannot be executed at the destination workstation.

The time parameters (**-t**, **-tb**, and **-ta**) are specified using the 24-hour clock. When you specify hours and minutes, use the format that is defined in your local system. To find out which format is defined, enter the **time** command at the command line and look at the output. For example, if the time includes a colon (:), you specify 2 a.m. and 4:37 p.m. as 2:00 and 16:37.

The date parameters (**-d**, **-db**, and **-db**) are specified using two digits each for the day, month, and year. When you specify the day, month, and year, use the format that is defined in your local system. To find out which format to use, enter the **date** command at the command line of your computer. For example, if the date includes a slash (/), and the month is before the day, you specify 10 July 2000 as 7/10/00.

Schedule Date and Time

The default values are:

- If you omit both **-d** and **-t**, the date and the time when you issued the command.
- If you specify **-d** but omit **-t**, the start of the day (just after midnight).
- If you specify **-t** but omit **-d**, the day when you issued the command.

Execution Date and Time

You can specify both execution and expiration date and time in terms of the *local* date and time at the origin or destination.

The execution date and time must be equal to or later than the scheduled date and time.

When the request is recursive you cannot specify **-db**.

If the recursion is specified as a time interval, the execution date and time cannot be specified at all.

The default values are as follows:

- The same rules as for **-d** and **-t**.
- If you specify an earlier time for **-tb** than for **-t** and you omit **-db**, then the date assumed by **-db** is the **-d** +1.

Expiration Date and Time

You can specify both expiration and execution date and time in terms of the *local* date and time at the origin or destination.

If you specify the **-da** and **-ta** parameters you cancel the request if it has not started by a certain date and time.

The expiration date and time must be later than the execution date and time, if both are specified.

When the request is recursive, you cannot specify **-da**.

If the recursion is specified as a time interval, the expiration date and time cannot be specified at all.

The default values are as follows:

- If you omit both **-da** and **-ta**, never is used as default.
- If you specify **-da**, but omit **-ta**, the default is the end of the day (just before midnight).
- If you specify **-ta**, but omit **-da**, the default day is the day specified by **-db**. If you do not specify any value for **-db**, the default value is applied.
- If you specify a lower value for **-ta** than for **-tb** and you omit **-da**, then the value assumed by **-da** is the value assumed by **-db** +1.

Recursion Expiration Date and Time

If you specify both a date and a time for the expiration of the recursion mechanism (**-ed** and **-et** respectively) the values must be later than **-d** and **-t**.

The default values are as follows:

- If you omit both **-et** and **-ed**, never.
- If you specify **-ed**, but omit **-et**, the end of the day (just before midnight).
- If you specify **-et**, but omit **-ed**, the day specified by **-d**.

Examples of Date and Time Usage

The following examples demonstrate the use of date and time parameters:

```
nvdmsend obj destination -w source -d 9/18/00 -t 18:00 -db 9/19/00 -tb 7:00
```

Command-Line Interface

TME 10 Software Distribution processes the obj file from the origin workstation on 9/18/00 at 6 p.m. and stores it to the destination workstation on 9/19/00 at 7 a.m. (date and time at origin workstation).

```
nvdmsend obj destination -w source -d 9/18/00 -t 18:00 -db 9/19/00 -tb 7:00 -z d
```

Unlike the previous example, the object is stored at the destination workstation when the local time at destination is 7 a.m. on 9/19/00. The **-z** parameter ensures that the operation takes place at the destination workstation at the date and time specified.

```
nvdmsend obj destination -w source -d 9/26/00 -t 18:00 -tb 7:00 -rd 0,1
```

Every month starting from 9/26/00, TME 10 Software Distribution processes the obj file from the origin workstation at 6 p.m. and stores it to the destination workstation at 7 a.m. of the next day.

```
nvdmsinst obj -w destination -d 9/18/00 -t 18:00 -db 9/19/00 -tb 4:00 -ta 8:00 -z d
```

The request is transferred to the owning server of the destination target at 6 p.m. on 9/18/00. TME 10 Software Distribution installs the obj file in the destination workstation on 9/19/00 from 4 to 8 a.m. (date and time at destination).

Recursive Parameters

When specifying recursive parameters, remember that:

- The **-rd** (recursion date interval), **-rt** (recursion time interval), **-rw** (recursion week schedule), and **-rm** (recursion month schedule) parameters are mutually exclusive.
- The **-rs** (recursion max severity), **-ed** (recursion expiration date), and **-et** (recursion expiration time), and **-ro** parameters can be specified only if one of the parameters listed in the first paragraph is specified.

Running a Procedure after a Request Has Completed

You can specify a cataloged procedure to be executed at a target after a request has completed. Use the **-m** target:procedure option to indicate the target and the procedure to be run. You can use this option if it is explicitly shown in the command syntax.

TME 10 Software Distribution executes the procedure using automatically the following parameters:

- The request ID, composed of four tokens
- The type of request
- The request status is a numeric value. It is defined in the `/usr/lpp/netviewdm/fndshrr.h` file.
- The severity level reached by the request

For a single request, the following parameters are also automatically used:

- Number of total addressed targets

- Number of targets at which the request completed
- Number of targets at which the request failed
- Number of targets at which the request was deleted

Return Codes and Error Messages

All commands set the return code as follows:

- Return code 0 means that the command succeeded
- Return code greater than 0 means that the command failed

The same return code can be issued by different commands but it always has the same meaning. The return code is 3 digits long and is the same as the numeric part of the message log entry that is generated.

Messages are issued to `stderr` when the command-line interface detects an error. They are identified by a unique code in the form:

FNDCL*aaab*

Where

aaa Is a number that identifies the message.
b Is the message severity, which can be either I, W, or E.

For more details about the individual messages that are displayed and how to interpret them, see the online *Message Reference*.

Input and Output Used for Command Scripts

The ways in which standard input and output are used are important if you are writing scripts that process TME 10 Software Distribution commands. Standard input, output, and error are used as follows:

- All command input is read from the command line. If additional input is required (for example, confirmation of a delete request), this is read from the `stdin` file.
- All errors are written to standard error.
- All other output is written to standard output.

This allows redirection to be used if required.

Language Variable

TME 10 Software Distribution is available in the English language only. If your operating system is not in English, this fact can cause some confusion, because prompts from the product in English do not correspond to actual responses that you must enter in the language in which your environment is installed.

For example, if you are working in Italian, and receive the following message in English:

```
uncatalog flat.data.res y/n?
```

you must respond in Italian using either:

Command-Line Interface

s (si) or n (no)

To ensure full message function, set the LANG environment variable to the language appropriate to your system, not to the value C.

Authorization Levels for Command Use

The following table shows the commands available at each authorization level.

<i>Table 4 (Page 1 of 2). Authorization Levels for Command Use</i>	
Authorization Level	Commands Allowed
All levels	help log svr
Configuration - View	lscm lsgp lstg
Configuration - Modify	addgp, lsgp, delgp addtg, lstg, deltg, updtg, addpm, delpm inv ls, lscf lscm, delcm rentg rld tron, troff
Preparation - View	ls, lscf
Preparation - Modify	cat, uncat, ls, bld, lscf, exp, imp
System Administration - View	lsbs lsusr stattg
System Administration - Modify	start, stop, reset hldtg, reltg updbbs, lsbs updusr, lsusr
Send - Yes	ls, send, cat
Retrieve - Yes	ls, rtrv
Queues - View	lsq stat
Queues - Modify	hlcd, relc lsq prgq, hldq, relq stat
Change Management Install - Yes	inst lscf lsrq, lscm

<i>Table 4 (Page 2 of 2). Authorization Levels for Command Use</i>	
Authorization Level	Commands Allowed
Change Management Execute - Yes	exec ls lscm
Change Management Authorize - Yes	auth, unauth ls lscm, delcm

Chapter 2. TME 10 Software Distribution Line Commands

This chapter contains an alphabetical list of all TME 10 Software Distribution commands.

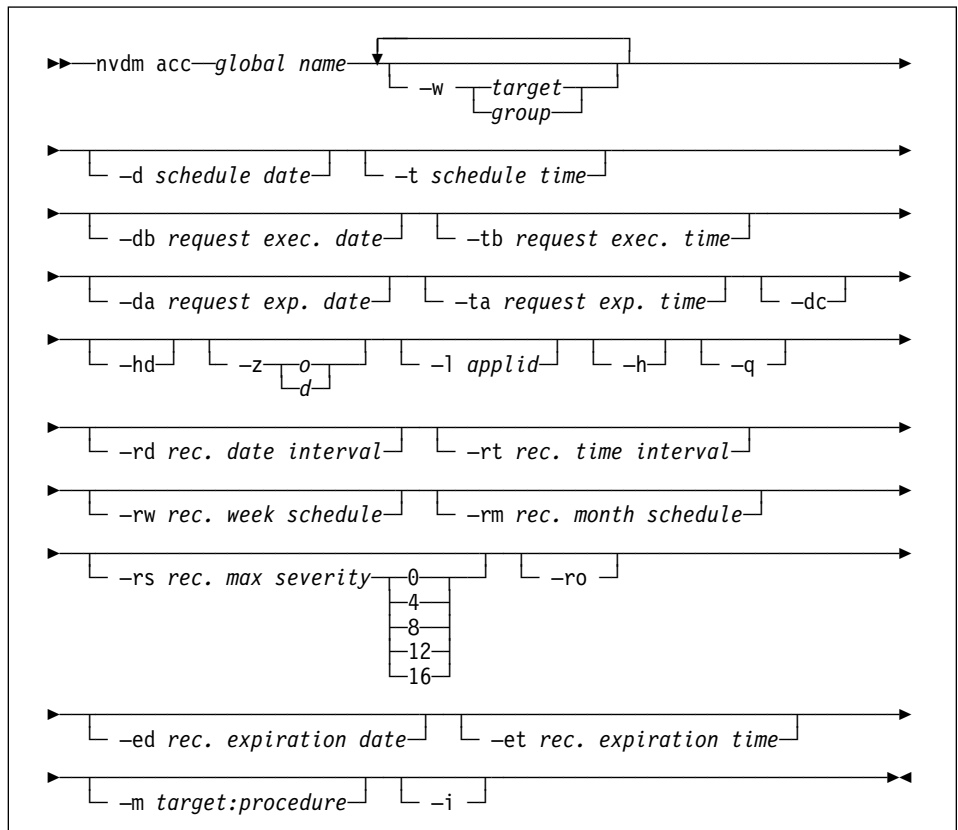
acc – Accept an Installed Change File

Use the **acc** command to accept a previously installed change file on a target. All backup copies of files that were changed by the installation are deleted. In order for the Accept command to be valid, the change file's status on the target must be:

- Installed, removable, active
- Installed, removable, inactive

This command can be used locally by mobile clients. It does not, however, accept the -w, -h, -z, -rd, -rt, -rw, -rm, -rs, -ro, -ed, -et or -m parameters.

Syntax



Parameters

global name

Global name of the change file to accept. See “Entering Global Names” on page 13.

You can use wildcard characters, but a unique match must be found.

-w *target | group*

The targets or groups to which this command applies. See “Entering Target Names” on page 12.

This parameter cannot be used locally by mobile clients.

-d *schedule date*

The date the request is to take place at the origin target.

Date and time parameters are described in more detail in “Entering Dates and Times” on page 13.

-t *schedule time*

The time the request is to take place at the origin target.

-db *request execution date*

The date the request is to take place at the destination target.

-tb *request execution time*

The time the request is to take place at the destination target.

-da *request expiration date*

The date after which the request cannot take place at the destination target.

-ta *request expiration time*

The time after which the request cannot take place at the destination target.

-dc

This parameter refers to operations performed at mobile clients. When it is specified, the operation is to be performed when the mobile client is disconnected from the server.

-hd

This parameter pertains to operations performed at mobile clients, and can only be specified together with the **-dc** parameter. It holds the disconnected request at the server until its execution time is reached. The request is forwarded to the client during the next connection window opened after the execution time specified, and executed there when the client is disconnected.

-z o|d

The format of the execution time. Enter one of the following values:

- o** The time is expressed as the local time at the origin target. This is the default.
- d** The time is expressed as the local time at the destination target.

This parameter cannot be used locally by mobile clients.

-l *application id*

The ID of the application used to submit the request. This ID can be used later to filter the requests submitted by a specific application.

It does not make sense to set an application ID for a request issued from the graphical interface, as this is an interactive process. Requests submitted from the

graphical interface are automatically given the value of GUI as the application ID. You cannot change this value.

If you do not specify an application ID for requests issued from the command-line interface, the default value is CLI.

- h** The request is to be submitted with high priority. You cannot specify this parameter if schedule times and execution times are specified.

This parameter cannot be used locally by mobile clients.

- q** The request is to be submitted in held state. The request is scheduled only when you release it.

This parameter cannot be used locally by mobile clients.

- rd** *rec. date interval*

A recursion date interval expressed in years, months, days. The request is executed upon every lapse of this interval. To specify an interval in terms of days, specify also the months and the years. To specify an interval in terms of months, specify also the years.

For example, 1 means the request is executed every year; 0,1 means the request is executed every month; 0,2,1 means the request is executed every day of every other month.

This parameter cannot be used locally by mobile clients.

- rt** *rec. time interval*

A recursion time interval expressed in hours and minutes. The request is executed upon every lapse of this interval. The format of a time interval is the same as for a time. For example, 3:45 means a time interval of 3 hours and 45 minutes.

This parameter cannot be used locally by mobile clients.

- rw** *rec. week schedule*

The days of the week when the recursive request is to take place. The days of the week are as follows:

0 = Sunday	4 = Thursday
1 = Monday	5 = Friday
2 = Tuesday	6 = Saturday
3 = Wednesday	

For example, to set a recursion for each Monday and Friday of the week, enter:

```
-rw 1,5
```

This parameter cannot be used locally by mobile clients.

- rm** *rec. month schedule*

The days of the month when the recursive request is to take place. Each day of the month is represented by a number from 1 to 31.

For example, to select a recursion for the 15th and 27th of the month, enter:

```
-rm 15,27
```

The number 31 specifies the last day of any month. The number 29 specifies the last day of February even in a non-leap year.

This parameter cannot be used locally by mobile clients.

-rs *rec. max severity* **0|4|8|12|16**

A new occurrence of the request is generated only if the severity level returned from the previous occurrence is less than or equal to the max-severity specified.

Specify one of the following:

- 0** Successful
- 4** Warning
- 8** Error
- 12** Failed
- 16** Unrecoverable

This parameter cannot be used locally by mobile clients.

-ro Specify this parameter to stop the recursion mechanism if the current occurrence of the request is in progress when the new occurrence should be generated. If you do not specify this parameter, one or more occurrences may be skipped until the current one is completed, but the recursion mechanism is not stopped. This parameter is not set by default.

This parameter cannot be used locally by mobile clients.

-ed *rec. expiration date*

The date after which the system stops the recursion mechanism.

This parameter cannot be used locally by mobile clients.

-et *rec. expiration time*

The time after which the system stops the recursion mechanism.

This parameter cannot be used locally by mobile clients.

-m *target:procedure*

A cataloged procedure to execute on a given target when the request is completed. See "Running a Procedure after a Request Has Completed" on page 16.

This parameter cannot be used locally by mobile clients.

-i Force the Accept command to be scheduled even if the change file history on the target does not allow it. However, the accept will not succeed if the change file history forbids it at the time when the accept is to be executed.

Examples

```
nvdms acc EURO.WORDPROC.UPD.2.3.US -w FREDSSWS -w JOHNSWS -db 12/15/00 -da 12/16/00
```

This command accepts the previously installed change file EURO.WORDPROC.UPD.2.3.US on targets FREDSSWS and JOHNSWS. The command is not executed before 15 December or after 16 December 2000.

acc

Related Information

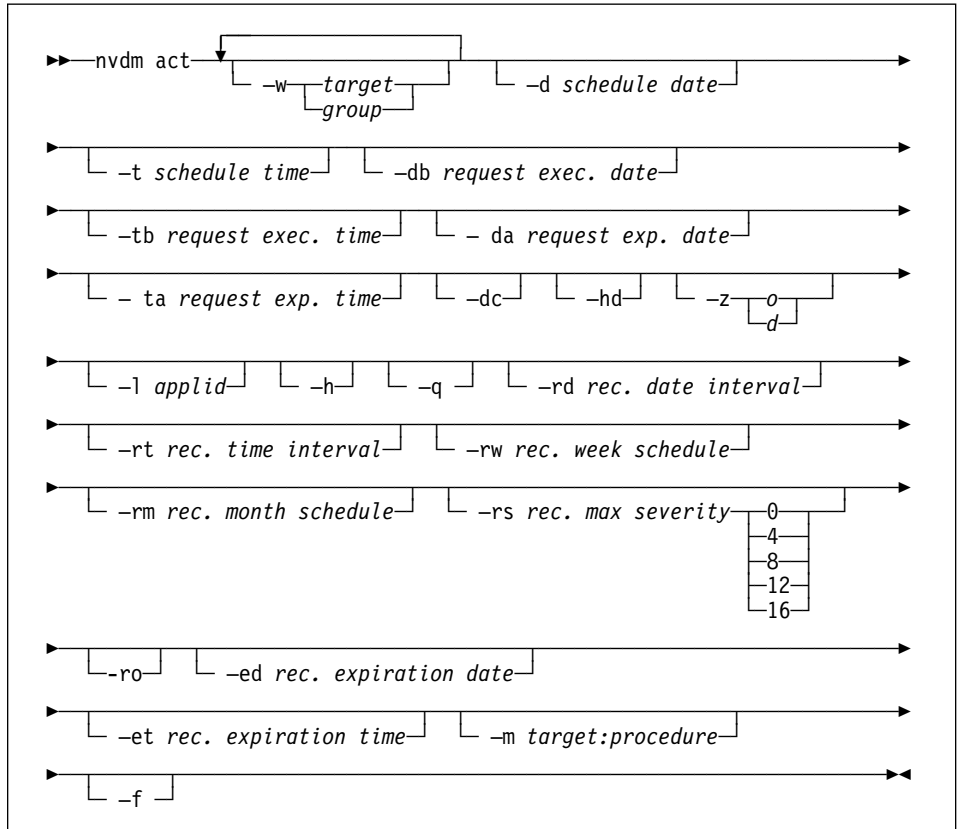
The lsrq, lscm, inst, rem, and uninstd commands.

act – Activate Changes on a Target

Use the **act** command to activate changes on a target.

This command can be used locally by mobile clients. It does not, however, accept the **-w**, **-h**, **-z**, **-rd**, **-rt**, **-rw**, **-rm**, **-rs**, **-ro**, **-ed**, **-et** or **-m** parameters.

Syntax



Parameters

-w *target | group*

The targets or groups to which the Activate operation applies. See “Entering Target Names” on page 12.

This parameter cannot be used locally by mobile clients.

-d *schedule date*

The date the request is to take place at the origin target.

Date and time parameters are described in more detail in “Entering Dates and Times” on page 13.

-t *schedule time*

The time the request is to take place at the origin target.

-db *request execution date*

The date the request is to take place at the destination target.

-tb *request execution time*

The time the request is to take place at the destination target.

-da *request expiration date*

The date after which the request cannot take place at the destination target.

-ta *request expiration time*

The time after which the request cannot take place at the destination target.

-dc

This parameter refers to operations performed at mobile clients. When it is specified, the operation is to be performed when the mobile client is disconnected from the server.

-hd

This parameter pertains to operations performed at mobile clients, and can only be specified together with the **-dc** parameter. It holds the disconnected request at the server until its execution time is reached. The request is forwarded to the client during the next connection window opened after the execution time specified, and executed there when the client is disconnected.

-z old

The format of the execution time. Enter one of the following values:

- o** The time is expressed as the local time at the origin target. This is the default.
- d** The time is expressed as the local time at the destination target.

This parameter cannot be used locally by mobile clients.

-l *application id*

The ID of the application used to submit the request. This ID can be used later to filter the requests submitted by a specific application.

It does not make sense to set an application ID for a request issued from the graphical interface, as this is an interactive process. Requests submitted from the graphical interface are automatically given the value of GUI as the application ID. You cannot change this value.

If you do not specify an application ID for requests issued from the command-line interface, the default value is CLI.

-h The request is to be submitted with high priority. You cannot specify this parameter if schedule times and execution times are specified.

This parameter cannot be used locally by mobile clients.

- q The request is to be submitted in held state. The request is scheduled only when released by the operator.

This parameter cannot be used locally by mobile clients.

-rd *rec. date interval*

A recursion date interval expressed in years, months, days. The request is executed upon every lapse of this interval. To specify an interval in terms of days, specify also the months and the years. To specify an interval in terms of months, specify also the years.

For example, 1 means the request is executed every year; 0,1 means the request is executed every month; 0,2,1 means the request is executed every day of every other month.

This parameter cannot be used locally by mobile clients.

-rt *rec. time interval*

A recursion time interval expressed in hours and minutes. The request is executed upon every lapse of this interval. The format of a time interval is the same as for a time. For example, 3:45 means a time interval of 3 hours and 45 minutes.

This parameter cannot be used locally by mobile clients.

-rw *rec. week schedule*

The days of the week when the recursive request is to take place. The days of the week are as follows:

0 = Sunday
1 = Monday
2 = Tuesday
3 = Wednesday
4 = Thursday
5 = Friday
6 = Saturday

For example, to set a recursion for each Monday and Friday of the week, enter:

```
-rw 1,5
```

This parameter cannot be used locally by mobile clients.

-rm *rec. month schedule*

The days of the month when the recursive request is to take place. Each day of the month is represented by a number from 1 to 31.

For example, to select a recursion for the 15th and 27th of the month, enter:

```
-rm 15,27
```

The number 31 specifies the last day of any month. The number 29 specifies the last day of February even in a non-leap year.

This parameter cannot be used locally by mobile clients.

-rs *rec. max severity 0|4|8|12|16*

A new occurrence of the request is generated only if the severity level returned from the previous occurrence is less than or equal to the max-severity specified.

act

Specify one of the following:

- 0** Successful
- 4** Warning
- 8** Error
- 12** Failed
- 16** Unrecoverable

This parameter cannot be used locally by mobile clients.

- ro** Specify this parameter to stop the recursion mechanism if the current occurrence of the request is in progress when the new occurrence should be generated. If you do not specify this parameter, one or more occurrences may be skipped until the current one is completed, but the recursion mechanism is not stopped. This parameter is not set by default.

This parameter cannot be used locally by mobile clients.

- ed** *rec. expiration date*

The date after which the system stops the recursion mechanism.

This parameter cannot be used locally by mobile clients.

- et** *rec. expiration time*

The time after which the system stops the recursion mechanism.

This parameter cannot be used locally by mobile clients.

- m** *target:procedure*

A cataloged procedure to execute on a given target when the request is completed. See "Running a Procedure after a Request Has Completed" on page 16.

This parameter cannot be used locally by mobile clients.

- f** Force the Activate operation to take place, even if the quiesce check shows that the target is still active.

The force parameter has no effect on a client.

Examples

```
nvdm act -w FREDWS -d 09/15/00 -t 03:00:00 -rw 1
```

This command schedules activation of all previously installed change files on target sales. The activation will take place as soon as possible after 3 a.m. on 15 September 2000 and will be executed every Monday. If FREDWS is not quiesced at the time the activation is to be performed, the activation will not take place.

Related Information

The inst, rem, uninst, and lsrq commands.

addgp – Add a Target Group

Use the **addgp** command to define a new group to the server, or to add up to nine targets to an existing group. You can only add static targets to existing dynamic groups. Groups can contain up to 1000 local and remote targets. The targets in a group do not necessarily have to be of the same mode.

You must have Modify Configuration authorization to use this command.

Syntax

```
nvdm addgp group id target -d description
```

— *-s short name* —

Parameters

group id

The name of the group. This name must be specified precisely, without using special characters.

target

Up to nine target names that are added to this group. Each target must be specified precisely, without using special characters. The target names must already have been added using the addtg command. If the group already exists, the targets are added to the group. If the group is dynamic, you can only add static targets to it. You can add a target to multiple groups by using the addgp command multiple times.

-d *description*

An optional description of the group being defined. If the group already exists, this description replaces the old one. Any character can be used. If the description contains blanks or any special characters that might be interpreted by the command processor, it must be enclosed by delimiters (usually double quotation marks).

If this parameter is omitted, no description is saved.

-s *short name*

The short name of the group. This parameter is mandatory when the group is first specified. If the command is being used to add targets to a group that already exists, this parameter is optional. If specified, it changes the short name of the group.

Examples

```
nvdm addgp WORD FREDWS JOHNSWS -d "Word Processor Users"
```

This command defines the group WORD, which contains two targets: FREDWS and JOHNSWS. Its description is Word Processor Users.

addgp

Related Information

The addpm, addtg, delgp, lsgp, and updtg commands.

addpln – Add a Plan

Use the **addpln** command to create a plan. The contents of the plan must be specified in a plan profile.

The following syntax checks are performed on a plan:

- Parameters of each entry in the plan profile must be correct. This includes the date and time parameters.
- All entry IDs in a condition must refer to an entry that is already defined in the plan profile.
- If the date and time parameters for an entry are specified as `local time at origin target`, any conditioning entry whose execution time is specified in the same format has to conform to this rule: the *not after* date and time of an entry must be later than the *not before* date and time of any conditioning entry.

All the options valid for a single request are allowed also in an entry of a plan profile.

Warning messages are issued when a request refers to any missing objects or targets.

Syntax

```

▶▶—nvdm addpln—profile name—┐—v┐▶▶

```

Parameters

profile name

Specifies the name of the file containing the plan profile.

-v Performs syntax checks on the plan profile without building the plan.

Examples

```
nvdm addpln euro.plan
```

This command builds a transmission plan from the plan profile `euro.plan`.

Related Information

The `execpln` command.

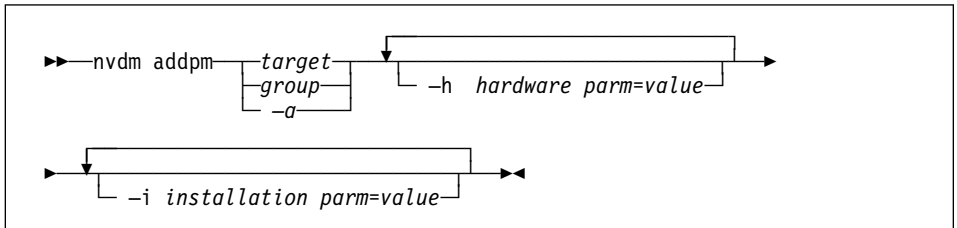
addpm – Add Parameters to a Target

Use the **addpm** command to add hardware and installation parameters to an existing target or group definition. All other aspects of the target or group configuration remain unchanged.

You can add parameters to all the targets in your domain by issuing the command once with the **-a** (all) parameter, or you can add parameters to specific targets only. Specific parameters are available to define, to all targets, a directory at the server where files are stored. This directory can then be mounted remotely at targets to access files to install.

You must have Modify Configuration authorization to use this command. It cannot be used for UI only targets.

Syntax



Parameters

All of the following parameters can be used for local targets. The **-h** parameter cannot be used for remote targets.

target | group | -a

Target or group to be modified. The name of the target or group must be precisely specified without the use of special characters.

Specify **-a** (all) when the installation parameters (**-i**) you are defining are shared by all the targets configured at this server. It allows you to specify the installation parameter only once for many targets, instead of for each target separately. You cannot specify **-a** when you are defining hardware parameters (**-h**).

The following options can each be repeated up to 10 times each. Each string should be enclosed by delimiters (usually double quotation marks) if it contains blanks or any special characters that might be interpreted by the command processor.

-h hardware parm=value

A hardware parameter to be used for checking hardware prerequisites when installing a change file. For example, **-h mem=40** can be used to check hardware prerequisites.

The **-h** parameter cannot be used for remote targets.

-i *installation parm=value*

An installation parameter to be used by the target during change file installation.

The following installation parameters can be defined as shared tokens (using the **-a** parameter) to mount remote directories from clients during the installation process:

SERVERREPOS

The name of the directory at the server to be exported.

SRVREPEXPOP

The options to be used to export the directory.

REMOTEREPOS

The name of the exported directory to be mounted at the target.

REMREPMNTOP

The options to be used for the mount process.

Examples

```
nvdm addpm FREDWS -i dir1=/usr/mydir
```

This command defines the parameter `dir1` at target `FREDWS`.

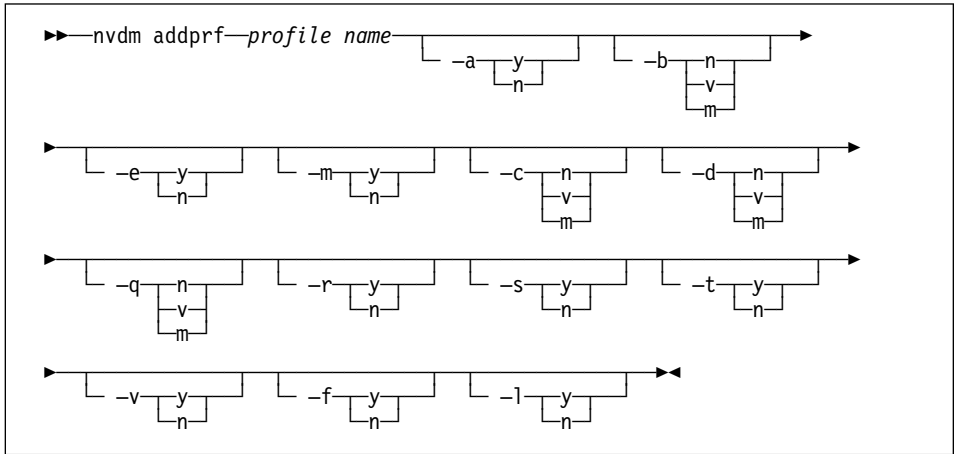
Related Information

The `delpm`, `addtg`, and `addgp` commands.

addprf – Add User Profile

Use the **addprf** command to create a new authorization profile. To be added, a profile must include at least one meaningful parameter.

Syntax



Parameters

If you omit one of the following parameters, the lowest authority level is assumed for it.

profile name

The name of the profile to be added.

-a y/n

Change management authorization operations. It allows access to the following functions:

- Authorize
- Unauthorize
- Delete History

Set this parameter for access to change management authorization to:

- y** Allow access.
- n** Deny access.

-b n|v|m

Build authorization. It can be set to one of the following:

- n** Deny access to build information.
- v** Allow information about the building of change files to be viewed.
- m** Allow all operations including the building of change files.

-e y/n

Change management execute authorization.

- y** Allow access.
- n** Deny access.

-m y|n

Change management operations. It allows access to the following functions:

- Install
- Remove
- Accept
- Uninstall

It can be set to one of the following values:

- y** Allow access.
- n** Deny access.

-c n|v|m

Configuration authorization. It can be set to one of the following values:

- n** Deny access to configuration information.
- v** Allow configuration information to be viewed.
- m** Allow all operations including the modification of configuration information.

-d n|v|m

System administration authorization. It can be set to one of the following values:

- n** Deny access to system administration information.
- v** Allow system administration information to be viewed.
- m** Allow all operations including modification of system administration information.

-q n|v|m

Queue operations. It can be set to one of the following values:

- n** Deny access to queue information.
- v** Allow queue information to be viewed.
- m** Allow all operations including queue management.

-r y|n

Retrieve authorization. It can be set to one of the following values:

- y** Allow access.
- n** Deny access

-s y|n

Send authorization. It can be set to one of the following values:

- y** Allow access.
- n** Deny access.

-t y|n

Target management authorization. It can be set to one of the following values:

- y** Allow change management at all targets.
- n** Allow to manage own target only.

addprf

-v y|n

Change management activation operation. It can be set to one of the following values:

- y** Allow access.
- n** Deny access.

-f y|n

Purge requests authorization. It can be set to one of the following values:

- y** Allow purge on requests.
- n** Do not allow purge on requests.

-l y|n

All requests flag. It can be set to one of the following values:

- y** Allow management of all requests.
- n** Allow management of own requests only.

Examples

```
nvdms addprf EUROSALLES -an -bn -en -my -cv -dn -qv -ry -sy -tn -vy -fn -ln
```

This command creates a user profile, EUROSALLES, to be associated with EuroTravel sales personnel. The users are authorized to perform the following functions at their own targets:

- Perform change management install, remove, accept and uninstall operations
- Perform send and retrieve operations
- Activate changes
- Manage their own requests

EUROSALLES users are *not* authorized to:

- Perform change management authorization operations
- Build change files
- Execute change management functions
- Perform system administration
- Purge requests

EUROSALLES users can view:

- Network configuration information
- Queues

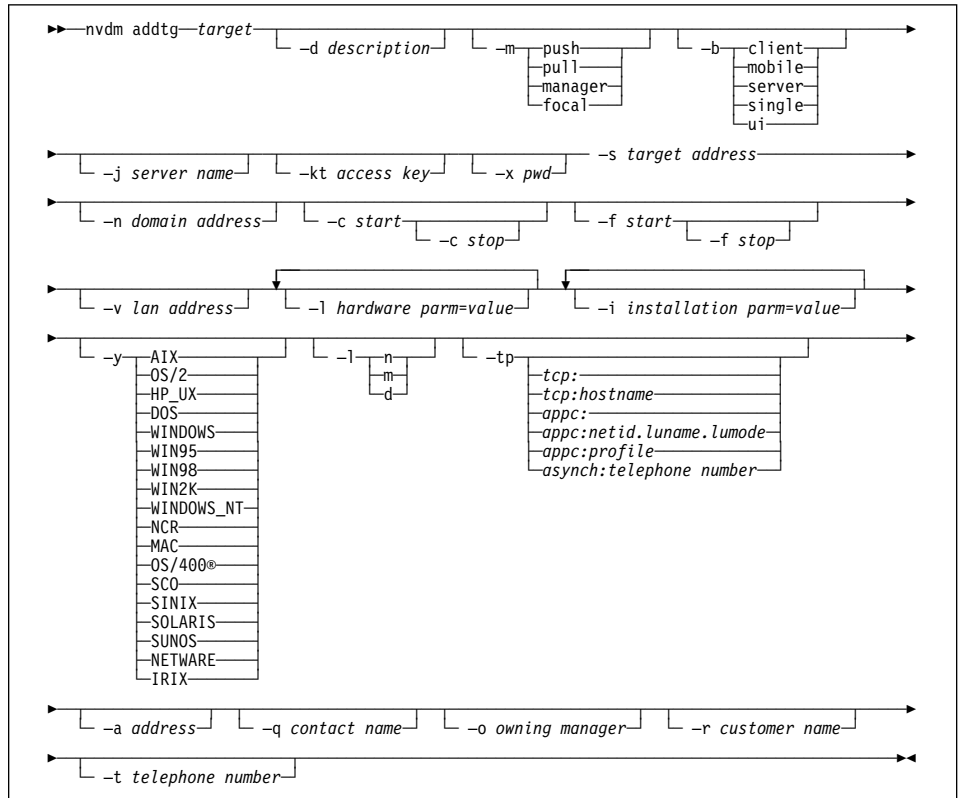
Related Information

The delprf and lsprf commands.

addtg – Create a Target

Use the **addtg** command to define a new target. You must have Modify Configuration authorization to use this command.

Syntax



Parameters

target

Target to be defined. The name must be precisely specified.

If you specify the name of an existing target, the command is rejected. Use the **uptdg** command to update an existing target definition. Alternatively, use the **deltg** command to delete a target definition before redefining it with this command.

-d *description*

A description of the target being defined. If the description contains blanks or any special characters that might be interpreted by the command processor, it must be enclosed by delimiters (usually double quotation marks). If this parameter is omitted, no description is saved.

-m push|pull|manager|focal

The mode in which this target can be used. The valid options are:

- push** Push mode target. Use this mode for client, server, and single target types. It is the default.
- pull** Pull mode target. Use this mode for client, server, and single target types.
- manager** The target is a manager, meaning that it can issue change control requests to any target in your domain. You can define more than one manager. Use this mode for server and single target types.
- focal** The target is a focal point, meaning that all change control reports (both local and remote) are to be sent here. A focal point is also a manager. You can define only one focal point. Use this mode for server and single target types.

-b client|server|single|ui |mobile

The type of target. The valid options are:

- client** The target is a client. If you specify a remote client, you must also specify the name of the server that the client is connected to using the **-j** option.
- server** The target is a server.
- single** The target is a single node.
- ui** The target is user interface only.
- mobile** The client has the mobile function installed.

-j server name

The server the client is connected to. Specify this parameter only if the target type is client. The default is the server the client is physically connected to.

-kt access key

The target access key. This parameter is optional. The default is none, meaning that no access key is assigned to the target. If you specify this parameter, the access key must exist, and must be assigned to users of this target.

-x pwd

The password to be used to access the target. The password is necessary only if the TARGET PASSWORD AUTHENTICATION keyword in the server base configuration file is set to YES. If it is, then whenever a target attempts communication with its server, the target sends it its password and the server checks to make sure that it corresponds to the password it has stored in the target configuration.

The password must be between 6 and 8 characters long and entered twice in the form `pwd/pwd`.

This parameter is optional. If you do not specify a value, the name of the target is used.

-s target address

The address for this target. For systems with remote communication, this parameter represents the second part of the TME 10 Software Distribution transport address. In a local network, the target address assigned to a server must also be used as the routing group name (RGN) for the server itself and all clients in its domain.

This field is required for all targets.

-n domain address

This parameter is valid only for remote targets. It can be up to 8 characters long; the only valid characters are letters and digits. The value you define with this parameter depends on the type of target you are defining. The following defaults exist for these target types:

client	The target address of the server the client is connected to.
server	The address of this target.
single	The address of this target.

-c start [-c stop]

The period of time allocated for change control requests. If you use the parameter twice, the first occurrence is the start time and the second is the stop time. If only one time is specified, it is assumed to be the start time; the stop time is the default.

The start time defaults to 00:00, and the stop time to 23:59 (that is, change control operations can be performed at any time).

You can specify this parameter only for local targets. It cannot be used for UI only, remote, or focal point targets.

-f start [-f stop]

The period of time allocated for distribution requests. If you use the parameter twice, the first occurrence is the start time and the second is the stop time. If only one time is specified, it is assumed to be the start time; the stop time is the default.

The start time defaults to 00:00, and the stop time to 23:59 (that is, distribution operations can be performed at any time).

You can specify this parameter only for local targets. It cannot be used for UI only, remote, or focal point targets.

-h hardware parm=value

The definition of a hardware parameter that is used for checking hardware prerequisites when installing a change file (for example `-h mem=40`). To redefine an existing parameter, you must first delete it using the `delpm` command.

You can specify this parameter up to 10 times for each command. You can define up to 128 hardware parameters for each target. You cannot use this parameter for UI only or focal point targets.

-v lan address

MAC network burned-in address for this target. The default is no address.

This parameter cannot be used for remote or focal point targets.

-i *installation parm = value*

An installation parameter to be used by the target during change file installation. You can specify this parameter up to 10 times for each command. It can be specified only for local targets. You cannot use this parameter for UI only, remote, or focal point targets.

To redefine an existing parameter, you must first delete it using the `delpm` command.

-y **AIX|OS/2|HP_UX|DOS|WINDOWS|WIN95|WIN98|WIN2K|WINDOWS_NT|SCO|NCR|MAC|SINIX|SOLARIS|SUNOS|NETWARE|IRIX|OS/400**

The operating system type for targets. Note that WINDOWS indicates Windows 3.11, while WIN95, WIN98 and WIN2K are Windows 95, Windows 98 and Windows 2000, respectively.

-l **n|m|d**

The level of message logging to be performed. The values are:

n Normal (the default)
m Minimal
d Diagnostic

You cannot specify this parameter for UI only or remote targets.

-tp *protocol type*

This parameter is applicable only when STS communication is being used. It specifies the type of communication protocol used by a client to connect to the server or by a server to connect to another server.

Valid options are:

tcp: The transmission protocol is TCP/IP.

tcp:hostname

The transmission protocol is TCP/IP, with the host name you specify.

appc: The transmission protocol is APPC.

appc:netid.luname.lumode

The transmission protocol is APPC, with the specified network ID, LU name, and LU mode related to the partner.

If you specify the parameter using this format, be sure to include all of these values.

appc:profile

The transmission protocol is APPC, with the name of the SNA Server/6000 Version 2.1 side destination profile you specify.

asynch:telephone number

The transmission protocol is asynchronous, which is used to link to a mobile client. Specify the telephone number used to connect to the client.

The default values are as follows:

- If you omit this parameter, **-tp tcp:** is the default.
- If **-tp tcp:** is specified, the target name is used.
- If **-tp appc:** is specified, the **-tp appc:netid.luname.lumode** option is used, where:

The network ID defaults to the target's routing group name (RGN)

The LU name defaults to the target's routing element name (REN)

The LU mode is blank

You can use this parameter for servers and local clients connected across STS.

-a *address*

Contact address for the specified target. The default is blank. Use an asterisk (*) in the address to begin a new line.

-q *contact name*

A contact name at the target. The default is blank.

-o *owning manager*

Owning manager of the target. The default is blank.

-r *customer name*

The name of the customer at the target. The default is blank.

-t *telephone number*

A telephone number at the target. The default is blank.

Examples

```
nvdn addtg FREDWS -m push -b client -j server01 -kt TAK2
-d "Fred's Computer" -s FRED -c 2100 -c 2330
-v 10:0:5a:3b:d3:41 -y AIX -tp tcp
```

This command defines the remote target FREDWS. It is a client configured in push mode and is connected to server01. The target address is FRED. Change control can be performed between 9 p.m. and 11:30 p.m. The burned-in MAC network address of the target is 10:00:5A:3B:D3:41 and the target runs the AIX operating system. It communicates with the server across a TCP/IP link.

```
nvdn addtg HOST -m focal -b server -d "The host">
-n NETWK1 -s LU0001 -tp appc:NETWK1.LU0001>
-r "Head Office" -q "Mr S.W. Shipper" -t "171 201 1212">
-a "The Eurotravel Corp.*Progress Boulevard*London"
```

This command defines the host processor at the head office. This is a remote target known as HOST. It uses an APPC communication protocol, and its RGN.REN name is NETWK1.LU0001.

addtg

Address details show Mr Shipper as the contact point and his telephone number.

Related Information

The addpm, deltg, delpm, lstg, rentg, stattg, and updtg commands.

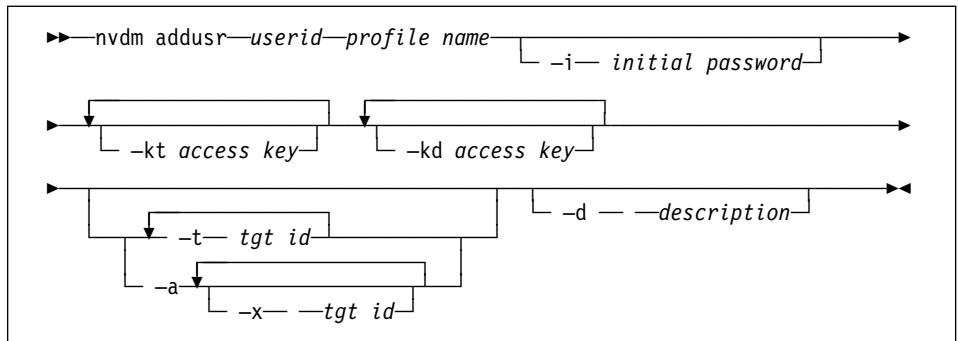
addusr – Create a User

Use the **addusr** command to create a new TME 10 Software Distribution user. When the product is installed, a user named **root** is created with the following defaults:

- Profile name = FNDADM
- No password is set
- -kt all
- -kd all
- -a

If you use this command to add a new user without specifying either **-t** or **-a**, the user is defined but cannot log on to any targets.

Syntax



Parameters

userid

The user's identifier. It can have up to eight alphanumeric characters, excluding all special characters.

profile name

The name of an existing authorization profile.

-i *initial password*

The password initially assigned to the user. The user can change it using the **updpwd** command. If you do not specify this parameter, no password is set: the user logs on specifying the *userid* only.

-kt *access key*

The target access key (TAK) associated with the user. If specified, the TAK name must exist in the access key table.

If you specify **all**, all access keys are associated with this user. The default is **none**, meaning no access keys are assigned to the user.

-kd *access key*

The data access key (DAK) associated with the user. If specified, the DAK name must exist in the access key table.

addusr

If you specify **all**, all access keys are associated with this user. The default is **none**, meaning no access keys are assigned to the user.

-t *tgt id*

The list of targets where the user is allowed to log in.

-a If this parameter is specified, the user is enabled to log in every target.

-x *tgt id*

The list of targets where the user is not enabled to log in.

-d *description*

A description of the user being defined. If the description contains blanks or any special characters that might be interpreted by the command processor, it must be enclosed by delimiters (usually double quotation marks). If this parameter is omitted, no description is saved. Enter up to 59 characters.

Examples

```
nvdms addusr JANEDOE EUROSALES -kt TAK2 -kd DAK2
```

This command adds the user JANEDOE. Her user authorization profile is EUROSALES, and she can access targets with the TAK2 access key and data with the DAK2 access key.

Related Information

The lsusr, updusr, delusr, addprf, lspfr, and updak commands.

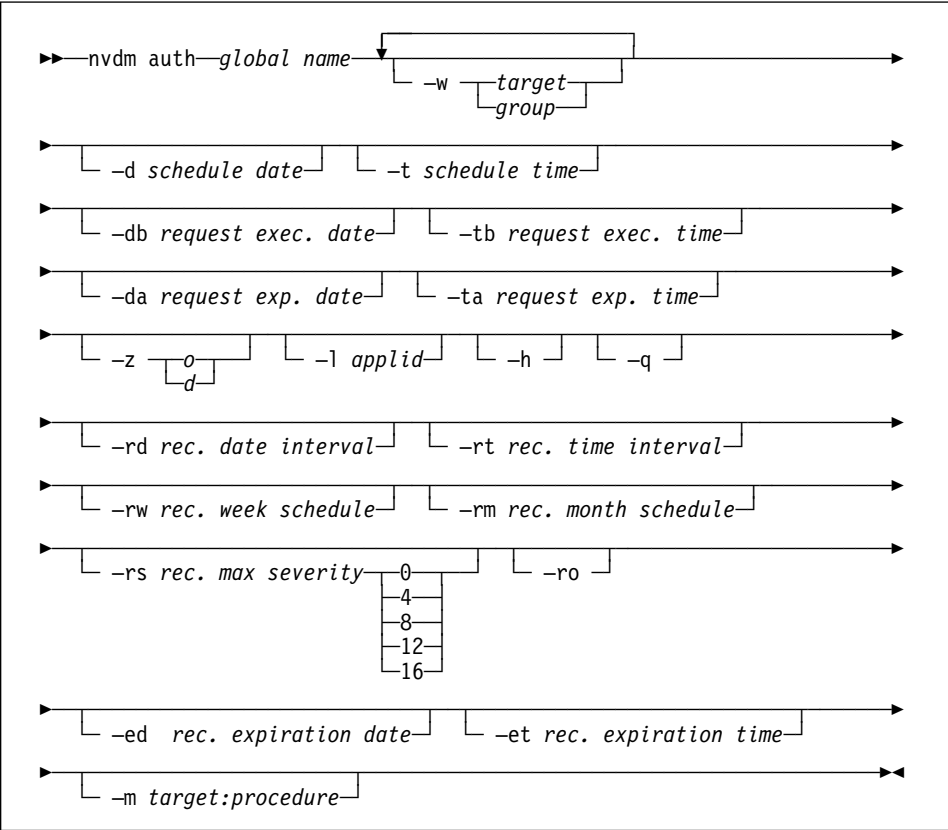
auth – Authorize Targets to Access Files

Use the **auth** command to authorize:

- A change file for installation on a target or group
- A data file for execution by a target or group

This command must be issued before a change file can be installed. It can be performed on remote targets only if the targets involved in the operation are connected by the server-to-server (STS) transmission protocol (see the *TME 10 Software Distribution Quick Beginnings* manual).

Syntax



Parameters

global name

Global name of the change file or program to authorize. See “Entering Global Names” on page 13.

You can use wildcard characters, but a unique match must be found.

-w *target | group*

The targets or groups to which this command applies. See “Entering Target Names” on page 12.

-d *schedule date*

The date the request is to take place at the origin target.

Date and time parameters are described in more detail in “Entering Dates and Times” on page 13.

-t *schedule time*

The time the request is to take place at the origin target.

-db *request execution date*

The date the request is to take place at the destination target.

-tb *request execution time*

The time the request is to take place at the destination target.

-da *request expiration date*

The date after which the request cannot take place at the destination target.

-ta *request expiration time*

The time after which the request cannot take place at the destination target.

-z *o|d*

The format of the execution time. Enter one of the following values:

- o** The time is expressed as the local time at the origin target. This is the default.
- d** The time is expressed as the local time at the destination target.

-h The request is to be submitted with high priority. You cannot specify this parameter if schedule times and execution times are specified.**-q** The request is to be submitted in held state. The request is scheduled only when released by the operator.

This parameter cannot be used locally by mobile clients.

-l *application id*

The ID of the application used to submit the request. This ID can be used later to filter the requests submitted by a specific application.

It does not make sense to set an application ID for a request issued from the graphical interface, as this is an interactive process. Requests submitted from the graphical interface are automatically given the value of GUI as the application ID. You cannot change this value.

If you do not specify an application ID for requests issued from the command-line interface, the default value is CLI.

-rd *rec. date interval*

A recursion date interval expressed in years, months, days. The request is executed upon every lapse of this interval. To specify an interval in terms of days,

specify also the months and the years. To specify an interval in terms of months, specify also the years.

For example, 1 means the request is executed every year; 0,1 means the request is executed every month; 0,2,1 means the request is executed every day of every other month.

-rt *rec. time interval*

A recursion time interval expressed in hours and minutes. The request is executed upon every lapse of this interval. The format of a time interval is the same as for a time. For example, 3:45 means a time interval of 3 hours and 45 minutes.

-rw *rec. week schedule*

The days of the week when the recursive request is to take place. The days of the week are as follows:

0 = Sunday
 1 = Monday
 2 = Tuesday
 3 = Wednesday
 4 = Thursday
 5 = Friday
 6 = Saturday

For example, to set a recursion for each Monday and Friday of the week, enter:

```
-rw 1,5
```

-rm *rec. month schedule*

The days of the month when the recursive request is to take place. Each day of the month is represented by a number from 1 to 31.

For example, to select a recursion for the 15th and 27th of the month, enter:

```
-rm 15,27
```

The number 31 specifies the last day of any month. The number 29 specifies the last day of February even in a non-leap year.

-rs *rec. max severity* **0|4|8|12|16**

A new occurrence of the request is generated only if the severity level returned from the previous occurrence is less than or equal to the max-severity specified.

Specify one of the following:

0 Successful
4 Warning
8 Error
12 Failed
16 Unrecoverable

-ro Specify this parameter to stop the recursion mechanism if the current occurrence of the request is in progress when the new occurrence should be generated. If you do not specify this parameter, one or more occurrences may be skipped until the

auth

current one is completed, but the recursion mechanism is not stopped. This parameter is not set by default.

-ed *rec. expiration date*

The date after which the system stops the recursion mechanism.

-et *rec. expiration time*

The time after which the system stops the recursion mechanism.

-m *target:procedure*

A cataloged procedure to execute on a given target when the request is completed.

Examples

```
nvdms auth EURO.WORDPROC.* -w FREDSSWS -d 06/01/00 -t  
6:00:00  
-db 06/01/00 -dt 7:30:00
```

This command authorizes FREDSSWS to use all files related to the Eurotravel word processing package. The command is executed at the origin workstation at 6:00 on June 1, 2000, and executed at the destination workstation at 7:30 on the same day. See “Running a Procedure after a Request Has Completed” on page 16.

Related Information

The inst, exec, and unauth commands.

bld – Build a Change File on the Server

Use the **bld** command to create a change file on the server . The contents of the change file are specified in the change file profile (see Chapter 3, “Creating Change File Profiles” on page 247). The new change file is cataloged as part of the build process. The build process always produces change files with object type software.

This command cannot be used at mobile clients when working with the local catalog.

Syntax

```

▶▶—nvdm bld—profile name—┐┐—┐┐—▶▶
                        └─f┐ └─gp┐

```

Parameters

profile name

The local name of the change file profile.

-f Forces an overwrite if you try to build a change file and the local file already exists on the server. If this option is not specified and a local file already exists on the server, you are asked to confirm an overwrite.

-gp

Imports product and filesset information from the source files of the change file. This parameter applies to AIX installp change files only.

Examples

```
nvdm bld /usr/mybuild/profile
```

This command builds, catalogs, and sends to the server the change file specified in the change file profile named `usr/mybuild/profile`.

Related Information

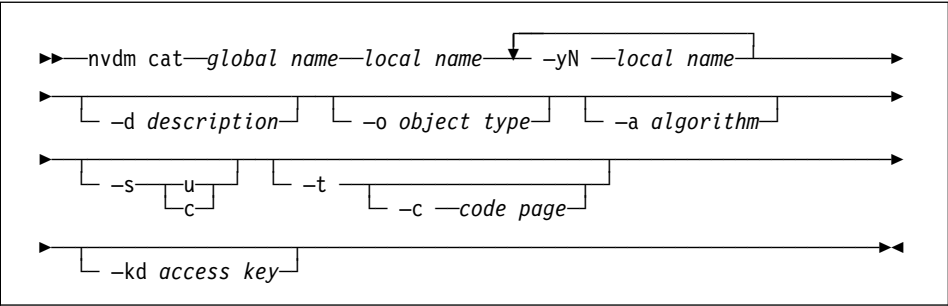
The `cat`, `lscf`, and `unbld` commands. Chapter 3, “Creating Change File Profiles” on page 247.

cat – Create a Catalog Entry

Use the **cat** command to create an entry in the catalog. Any file that is to be either sent or retrieved (or manipulated in any other way) must be cataloged. A file can be cataloged before it is received.

This command can be used locally by mobile clients.

Syntax



Parameters

global name

The global name of the file. You cannot use wildcard characters and you cannot define a global name that commences with the tokens `$DELETE.$PENDING`, as it is a reserved value.

local name

Specifies the local name assigned to file system ID 0 (zero).

-yN *local name*

Specifies additional local names of the file at each destination target, depending on target's file system. **N** indicates the specific file system ID the local file name refers to. You can specify up to seven different local file names associated with each file system ID.

The file system ID associated with every workstation is defined in its base configuration file (`nvdm.cfg`). Enter a value from 1 to 7 for **N**. You can specify up to seven file system IDs (together with the local name) for each file that you are cataloging. Enter a value from 1 to 7 for **N**.

Each platform picks the correct local name on the basis of its file system ID. For example, if you define file system ID=5 in the base configuration file of all OS/2 workstations, when you catalog a file to be sent to these workstations you have to associate the value 5 with the OS/2 local name.

If you do not define the file system ID in the base configuration file of the OS/2 workstations, when you catalog a file for OS/2 you must associate the default file system ID value, (which for OS/2 is 2), with the OS/2 local name. With this value the file that you send is stored in the `c:\usr\lpp\netviewdm\repos` directory. The following table lists the file system ID default values for the various platforms:

Table 5. Default File System IDs

Platform	File System ID	File Name Syntax
AIX/6000®	1	/usr/lpp/netviewdm/repos...
DOS, OS/2, Windows 3.11	2	c:\softdist\repos...
NetWare	3	sys:\system\ibmnvdm\repos...
Windows 95, Windows 98, Windows 2000, Windows NT	5	c:\softdist\repos...

If you do not specify the file system ID value for the OS/2 workstations, the default local name follows the syntax rules associated with file system ID = 0. If you do not define the local name associated with 0, the default local name follows the syntax of the source workstation.

You can catalog a change file specifying different file system IDs, but all change control operations use the value 0.

For example, if a file must be sent to two targets that have different file system IDs, specify:

```
nvdms cat nvdms.new.file localname0 -y1 localname1 -y4 localname4
```

-d *description*

An optional description of the entry being created. Any characters can be used. If the description contains blanks or any special characters that might be interpreted by the command processor, it must be enclosed in double quotation marks.

If this parameter is omitted, the description is left blank.

-o *object type*

The type of object being cataloged. The possible values, with allowed abbreviations, are in the following table. You can also represent the object type using its SNA/FS class code. Specify 0x followed by the corresponding 8 hexadecimal characters in C language format.

Table 6. Object Types and their SNA/FS Class Codes	
Object Type	Hexadecimal value
FLATDATA (FLATD)	0x20100000
SOFTWARE (SOFTW)	0x10300000
MICROCODE (MICR)	0x10100000
PROCEDURE (PROC)	0x10500000
RELDATA (RELD)	0x20200000
DUMP	0x40100000
CONFIGFILE (CONFIG)	0x40200000
TRACE	0x40300000
ERRLOG	0x40400000
PLAN	0x00801050

The default object type is FLATDATA.

If you are cataloging a data file and the global name contains REF, UPD, or FIX, the object type must not be FLATDATA, SOFTWARE, or MICROCODE. If any of these names are used, the file is treated as a change file.

-a *algorithm*

The packing (compression) algorithm that has been used on the cataloged file, or that will be used if the file is compressed in the future (for example, when distributing). The possible values of the compression algorithm are:

- **SNA** (SNA run-length encoding)
- **LZW** (modified PKZIP)
- *filename [parameters]*

where *filename* is the name of the file that contains a user-supplied algorithm, which should be present on all remote nodes to which the file might be sent. The file name can be up to 12 characters long, and must be followed by any parameters required by the compression algorithm. Specify up to a maximum of 27 characters.

If this parameter is omitted, the **LZW** algorithm is used if the file ever needs to be compressed.

-s u|c

The current compression state of the file:

- u** Uncompressed
- c** Compressed

If **-s** is omitted, the default compression type is uncompressed.

- t** The file to catalog is a text file. If you do not specify this parameter, the file is assumed to be binary.

-c *code page*

The code page of the text file. The code pages allowed are those that were defined on the workstation this configuration was installed on. If this parameter is omitted, the local system code page is used.

-kd *access key*

The data access key (DAK) associated with the file. If specified, the DAK name must exist in the access key table, and be associated with the user issuing the command. If not specified, no access key is assigned to the catalog entry.

Examples

```
nvdms cat EURO.SPELLCHECK.DAT.3.US /usr/amc/spellcheck -d
"The data file for use by the spell check utility (level 3, US)"
-o FLATDATA -a "megafast -z" -sc -t -c IBM-850
```

This command creates a catalog entry for the global name EURO.SPELLCHECK.DAT.3.US. It is a text file (with the code page IBM-850) that corresponds to the local file name /usr/amc/spellcheck. The description The data file for use by the spell check

utility (level 3, US) is included in the catalog. The file is currently compressed using the megarast compression algorithm with the **-z** parameter.

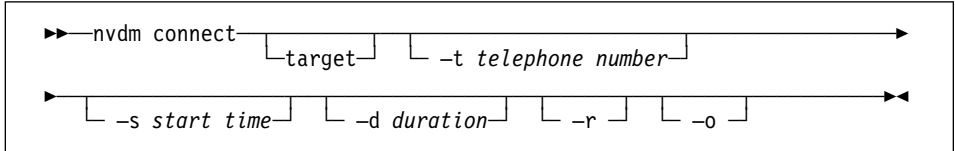
Related Information

The uncat command.

connect – Establish Mobile Connection with Server

Use the **connect** command to connect to a server from a mobile client.

Syntax



Parameters

<target>

The name of the client the connection is being set up for. This parameter is optional. If it is omitted, the name of the client issuing the command is assumed.

-t *telephone number*

The telephone number at which the client can be contacted by the server. It is a Hayes-compatible AT dial string, for example 441813639363,,374 (where commas denote pauses). The number is appended to the dial string configured for whichever port the server uses when returning the client's call.

If -t is not specified, the server uses the last telephone number used.

-s *start time*

The time when the connection window can be opened. The **start time** gives the local time at which the server can start trying to dial back to the client. The connection window remains open from the start time for the number of minutes specified in **-d duration** specified below.

-d *duration*

The length of time, expressed in minutes, that the connection window is to remain open. After the time has expired, the server can no longer attempt to call the client. The maximum window size is 1439 minutes.

If no parameters are specified, the defaults are:

- start time=current time
- duration=connection duration

The **connection duration** is the time you specified with the correspond keyword in the `nvdm.cfg` file. See the *TME 10 Software Distribution for OS/2 Quick Beginnings* for more details on the `nvdm.cfg` file.

- r Establishes a recursive daily connection from the client to the server during a specific connection window. The connection window is opened through the **-s start time** optional parameter, and it remains open from the start time for the number of minutes specified in **-d duration**. The recursion starts the day on which you submit the command. To close the recursion, submit the **disconnect target -r** command.

To change the recursion you previously set, resubmit the `connect -r` command specifying the new values. To establish a new connection that is not recursive, you must first close the recursion.

- o Manages outstanding requests. Specify this parameter only if you specified the -r optional parameter. The connection window is opened only if there are pending requests for the target. The outstanding requests are managed immediately if they arrive in the specified connection window. If the outstanding requests arrive after the connection window is closed, they are managed the next day.

del – Delete a File

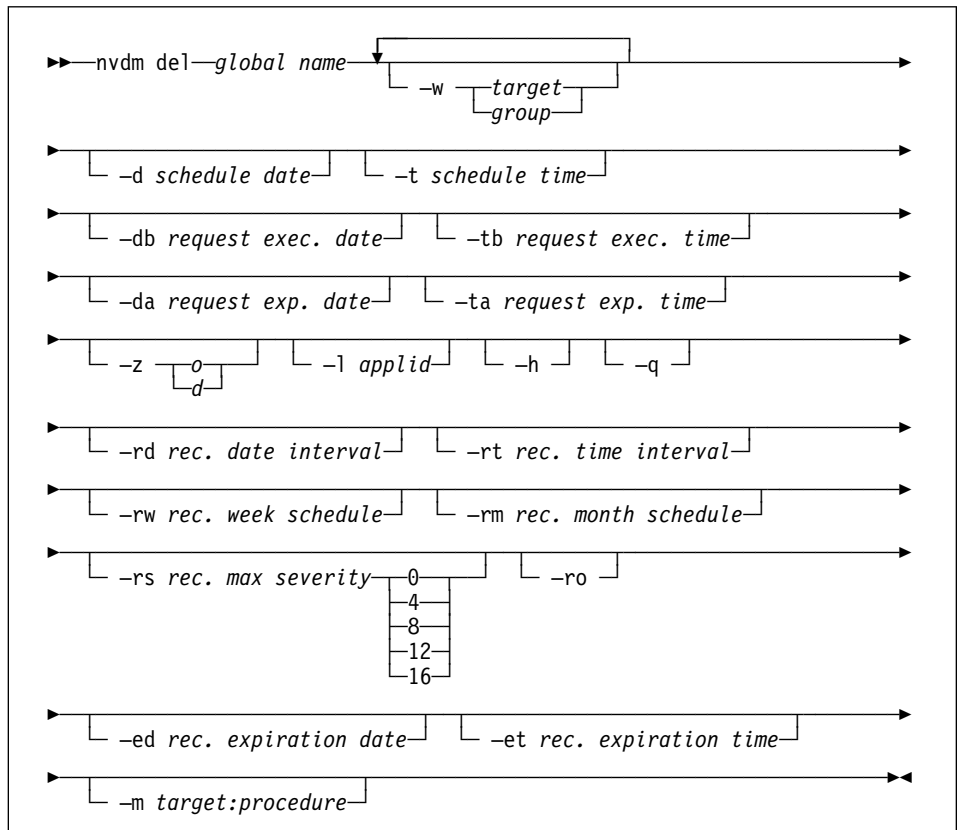
Use the **del** command to erase a file from a local or remote target or group. The file is identified by its global name. You can initiate delete operations on remote targets or groups only from a remote administration site.

Using this command to delete files only if they have been cataloged.

If the delete command deletes the last instance of the file in the domain, its catalog entry is automatically deleted. At the remote administration site this does not apply, because this site does not maintain a history of files at the various sites in the network.

This command cannot be used at mobile clients when working with the local catalog.

Syntax



Parameters

global name

The global name of the file to be deleted. See “Entering Global Names” on page 13.

You can use wildcard characters, but a unique match must be found.

-w *target | group*

The targets or groups where the file is to be deleted. See “Entering Target Names” on page 12.

-d *schedule date*

The date the request is to take place at the origin target.

Date and time parameters are described in more detail in “Entering Dates and Times” on page 13.

-t *schedule time*

The time the request is to take place at the origin target.

-db *request execution date*

The date the request is to take place at the destination target.

-tb *request execution time*

The time the request is to take place at the destination target.

-da *request expiration date*

The date after which the request cannot take place at the destination target.

-ta *request expiration time*

The time after which the request cannot take place at the destination target.

-z *old*

The format of the execution time. Enter one of the following values:

o The time is expressed as the local time at the origin target. This is the default.

d The time is expressed as the local time at the destination target.

-h The request is to be submitted with high priority. You cannot specify this parameter if schedule times and execution times are specified.

-q The request is to be submitted in held state. The request is scheduled only when released by the operator.

This parameter cannot be used locally by mobile clients.

-l *application id*

The ID of the application used to submit the request. This ID can be used later to filter the requests submitted by a specific application.

It does not make sense to set an application ID for a request issued from the graphical interface, as this is an interactive process. Requests submitted from the graphical interface are automatically given the value of GUI as the application ID. You cannot change this value.

If you do not specify an application ID for requests issued from the command-line interface, the default value is CLI.

-rd *rec. date interval*

A recursion date interval expressed in years, months, days. The request is executed upon every lapse of this interval. To specify an interval in terms of days, specify also the months and the years. To specify an interval in terms of months, specify also the years.

For example, 1 means the request is executed every year; 0,1 means the request is executed every month; 0,2,1 means the request is executed every day of every other month.

-rt *rec. time interval*

A recursion time interval expressed in hours and minutes. The request is executed upon every lapse of this interval. The format of a time interval is the same as for a time. For example, 3:45 means a time interval of 3 hours and 45 minutes.

-rw *rec. week schedule*

The days of the week when the recursive request is to take place. The days of the week are as follows:

0 = Sunday	4=Thursday
1 = Monday	5=Friday
2 = Tuesday	6=Saturday
3 = Wednesday	

For example, to set a recursion for each Monday and Friday of the week, enter:

```
-rw 1,5
```

-rm *rec. month schedule*

The days of the month when the recursive request is to take place. Each day of the month is represented by a number from 1 to 31.

For example, to select a recursion for the 15th and 27th of the month, enter:

```
-rm 15,27
```

The number 31 specifies the last day of any month. The number 29 specifies the last day of February even in a non-leap year.

-rs *rec. max severity* **0|4|8|12|16**

A new occurrence of the request is generated only if the severity level returned from the previous occurrence is less than or equal to the max-severity specified.

Specify one of the following:

0	Successful
4	Warning
8	Error
12	Failed
16	Unrecoverable

-ro Specify this parameter to stop the recursion mechanism if the current occurrence of the request is in progress when the new occurrence should be generated. If you do not specify this parameter, one or more occurrences may be skipped until the

current one is completed, but the recursion mechanism is not stopped. This parameter is not set by default.

-ed *rec. expiration date*

The date after which the system stops the recursion mechanism.

-et *rec. expiration time*

The time after which the system stops the recursion mechanism.

-m *target:procedure*

A cataloged procedure to execute on a given target when the request is completed. See “Running a Procedure after a Request Has Completed” on page 16.

Examples

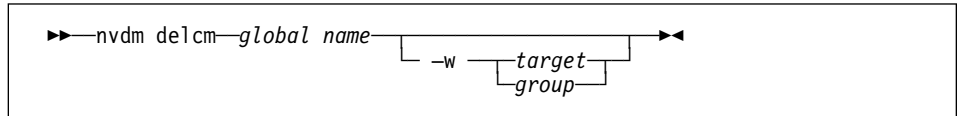
```
nvdm del MYBUILD.FIX -w FREDWS
```

This command deletes the local file whose global name is MYBUILD.FIX from the target FREDWS.

delcm – Delete a Change File Status Record

Use the **delcm** command to delete the change file status record holding the status of the given change file on the target or group.

Syntax



Parameters

global name

The global name of the change file whose status is to be deleted; see “Entering Global Names” on page 13.

You can use wildcard characters, but a unique match must be found.

-w *target | group*

The targets or groups to which this command applies.

You cannot specify more than one target or group at a time. You can use an asterisk (*) enclosed in single or double quotation marks to indicate all targets. See “Entering Target Names” on page 12.

If **-w** is not specified, your workstation is used as the default target.

Examples

```
nvdm delcm EURO.WORDPROC.REF.1.US -w “**”
```

This command deletes the change file status for EURO.WORDPROC.REF.1.US for all targets.

Related Information

The **lscm** command.

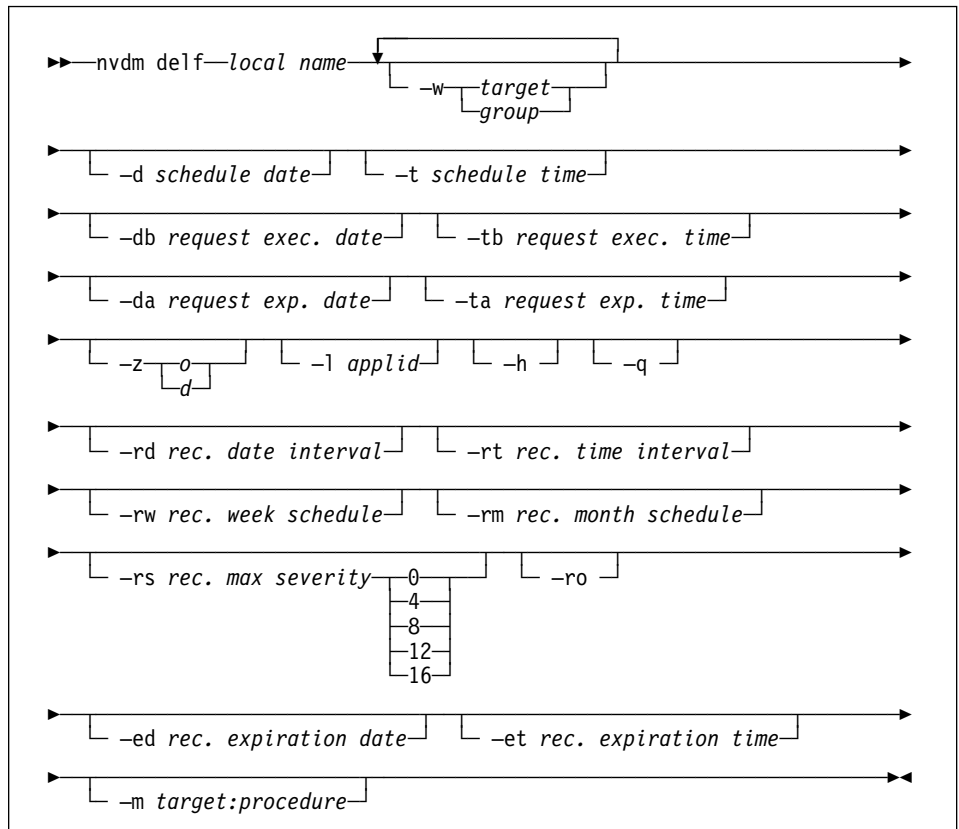
delf – Delete a Noncataloged File

Use the **delf** command to delete a file that is not cataloged from one or more targets. You identify the file to be deleted by its local name.

This operation can only be performed between targets connected by the server-to-server (STS) transmission protocol (see the *TME 10 Software Distribution Quick Beginnings* manual).

This command cannot be used at mobile clients when working with the local catalog.

Syntax



Parameters

local name

The local name of the file to be deleted. This parameter is required.

-w *target | group*

The targets or groups where the file is to be deleted. See “Entering Target Names” on page 12.

-d *schedule date*

The date the request is to take place at the origin target.

Date and time parameters are described in more detail in “Entering Dates and Times” on page 13.

-t *schedule time*

The time the request is to take place at the origin target.

-db *request execution date*

The date the request is to take place at the destination target.

-tb *request execution time*

The time the request is to take place at the destination target.

-da *request expiration date*

The date after which the request cannot take place at the destination target.

-ta *request expiration time*

The time after which the request cannot take place at the destination target.

-z o|d

The format of the execution time. Enter one of the following values:

o The time is expressed as the local time at the origin target. This is the default.

d The time is expressed as the local time at the destination target.

-h The request is to be submitted with high priority. You cannot specify this parameter if schedule times and execution times are specified.

-q The request is to be submitted in held state. The request is scheduled only when released by the operator.

This parameter cannot be used locally by mobile clients.

-l *application id*

The ID of the application used to submit the request. This ID can be used later to filter the requests submitted by a specific application.

It does not make sense to set an application ID for a request issued from the graphical interface, as this is an interactive process. Requests submitted from the graphical interface are automatically given the value of GUI as the application ID. You cannot change this value.

If you do not specify an application ID for requests issued from the command-line interface, the default value is CLI.

-rd *rec. date interval*

A recursion date interval expressed in years, months, days. The request is executed upon every lapse of this interval. To specify an interval in terms of days, specify also the months and the years. To specify an interval in terms of months, specify also the years.

For example, 1 means the request is executed every year; 0,1 means the request is executed every month; 0,2,1 means the request is executed every day of every other month.

-rt *rec. time interval*

A recursion time interval expressed in hours and minutes. The request is executed upon every lapse of this interval. The format of a time interval is the same as for a time. For example, 3:45 means a time interval of 3 hours and 45 minutes.

-rw *rec. week schedule*

The days of the week when the recursive request is to take place. The days of the week are as follows:

0 = Sunday
 1 = Monday
 2 = Tuesday
 3 = Wednesday
 4 = Thursday
 5 = Friday
 6 = Saturday

For example, to set a recursion for each Monday and Friday of the week, enter:

```
-rw 1,5
```

-rm *rec. month schedule*

The days of the month when the recursive request is to take place. Each day of the month is represented by a number from 1 to 31.

For example, to select a recursion for the 15th and 27th of the month, enter:

```
-rm 15,27
```

The number 31 specifies the last day of any month. The number 29 specifies the last day of February even in a non-leap year.

-rs *rec. max severity 0|4|8|12|16*

A new occurrence of the request is generated only if the severity level returned from the previous occurrence is less than or equal to the max-severity specified.

Specify one of the following:

0 Successful
4 Warning
8 Error
12 Failed
16 Unrecoverable

-ro Specify this parameter to stop the recursion mechanism if the current occurrence of the request is in progress when the new occurrence should be generated. If you do not specify this parameter, one or more occurrences may be skipped until the current one is completed, but the recursion mechanism is not stopped. This parameter is not set by default.

delf

-ed *rec. expiration date*

The date after which the system stops the recursion mechanism.

-et *rec. expiration time*

The time after which the system stops the recursion mechanism.

-m *target:procedure*

A cataloged procedure to execute on a given target when the request is completed.

See “Running a Procedure after a Request Has Completed” on page 16.

Examples

```
nvdms delf FILE1.TXT -w target2 -w target3 -w group1
```

This command causes the file FILE1.TXT to be deleted at target2, target3, and at all the targets belonging to group1.

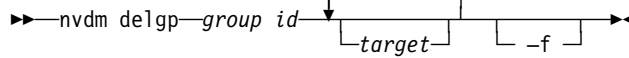
delgp – Delete a Target Group

Use the **delgp** command to delete a group definition from the network. The definitions of targets within that group are unaffected. It can also be used to remove a target from a group without affecting the other targets in the group. If a group is dynamic, you can only use the command to delete static targets from it.

You must have Modify Configuration authorization to use this command.

This command cannot be used in single-node systems, because a single-node system cannot have groups.

Syntax



```
nvdm delgp group id target -f
```

Parameters

group id

The name of the group to which this command applies. The group must be fully specified without using special characters.

target

Up to nine target names to be deleted from this group. Specify each target precisely, and ensure that it matches the name of a target that is in the group. If a group is dynamic, you can only delete static targets from it.

If no target is specified, the entire group is deleted.

If one or more of the targets is specified incorrectly or is not a member of the group, the targets are still deleted from the group.

If the command results in the last target being removed from the group, the group itself is not removed. You must run the command specifying no targets to delete the group.

-f Confirmation is not requested before the deletion is performed. If **-f** is omitted, you must confirm the deletion.

If you type Y, the deletion is carried out; otherwise, it is not.

Examples

```
nvdm delgp WORD FREDSW S HARRYSW S
```

This command removes the targets FREDSW S and HARRYSW S from the group WORD. No confirmation is required.

delgp

Related Information

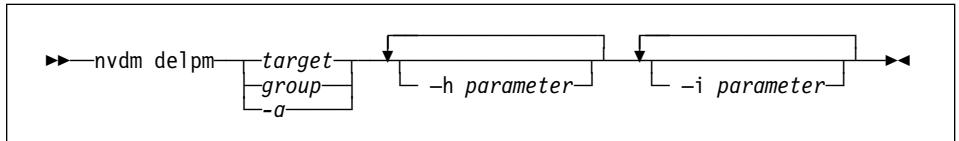
The addgp and lsgp commands.

delpm – Delete Parameters from a Target

Use the **delpm** command to delete hardware parameters or software parameters from a target or a group definition. You must have full configuration authorization to use this command. If you have installed a single-node system, you cannot use this command to apply to groups because there are no groups in your system.

This command cannot be used for UI only targets.

Syntax



Parameters

All of the following parameters can be used for local targets. The **-h** parameter cannot be used for remote targets.

target | group | -a

Target or group to be modified specified without special characters.

Specify **-a** (all) when the installation parameters (**-i**) you are defining are shared by all the targets configured at this server. It allows you to specify the installation parameter only once for many targets, instead of for each target separately. You cannot specify **-a** when you are defining hardware parameters (**-h**).

The following parameters can each be repeated up to 10 times. Each string should be enclosed by delimiters (usually double quotation marks) if it contains blanks or any special characters that might be interpreted by the command processor.

-h parameter

The definition of a hardware parameter to be deleted. This parameter cannot be used for remote targets.

-i parameter

The definition of an installation parameter to be deleted.

Examples

```
nvdm delpm FREDWS -i base -h mem
```

This command deletes two items from the configuration of target **FREDWS**: the installation parameter **base** and the hardware parameter **mem**.

Related Information

The **addpm** command.

delprf – Delete an Authorization Profile

Use the **delprf** command to erase an authorization profile.

Syntax

```
nvdm delprf profile name [-f]
```

Parameters

profile name

The name of the profile.

-f Force the deletion of the profile without requesting confirmation.

The system validates that:

- The profile name exists
- The profile name is not one of the default profiles
- The profile is not associated with any user

Examples

```
nvdm delprf EUROSALES -f
```

This command deletes the authorization profile EUROSALES. No confirmation is requested from the user before the deletion is performed.

Related Information

The addprf, lsprf, and updprf commands.

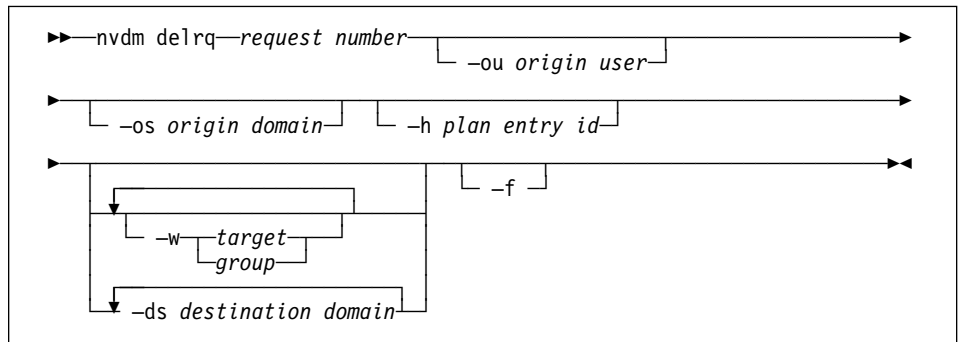
delrq – Delete a Request

Use the **delrq** command to delete:

- A request addressed to a specific target or targets
- A request to all targets addressed in a domain
- All requests in a plan

If authorized, you can delete remote requests too, provided that your network is connected across server-to-server (STS) connection.

Syntax



Parameters

request number

The sequential number of the request to be deleted. This parameter is required.

-ou *origin user*

The user who submitted the request. Only the administrator can delete requests submitted by other users. This parameter is optional. The default is the name of the user submitting this command.

-os *origin domain*

The domain the request was submitted from. This parameter is optional. The default is the local domain.

-h *plan entry id*

The identification of the plan entry to be deleted. If the request does not refer to a plan, an error message is displayed. All entries that are conditioned by this entry are deleted as well, even if their execution could still be possible. If you do not specify this parameter and the request refers to a plan, the entire plan is deleted. If you do not specify either **-ds** or **-w**, all requests for this entry are deleted.

-ds *destination domain*

The domain for which the request is to be deleted. This parameter cannot be specified if **-w** is specified.

delrq

-w *target | group*

The targets or groups where the request is to be deleted. See “Entering Target Names” on page 12.

This parameter cannot be specified if **-ds** is specified.

- f** Confirmation is not requested before the deletion is performed. If **-f** is omitted, you must confirm the deletion. When a plan entry is being deleted, all conditioned entries that are deleted along with it are displayed.

Examples

```
nvdm delrq 10 -ou fred -h 003
```

This command deletes entry 003 which belongs to request 10. The request was submitted by user FRED. The command was issued by the administrator, who has the authorization to delete requests submitted by other users.

Related Information

The eraserq, lsrq, relrq, and rstrq commands.

deltg – Delete a Target

Use the **deltg** command to delete a target configuration from the product database. The target is removed from any groups to which it belongs and the target parameters are also deleted from the product database.

The deleted target is removed from the list of targets accessed by a user. If that target was the only one accessed by a user, a warning message is logged for that user at the server.

You cannot delete a target if any requests are scheduled for it.

You cannot delete the server target, if any clients are defined to it.

You must have Modify Configuration authorization to use this command.

Syntax

```
►►—nvdm deltg—target—┐└─f─┐►◄
```

Parameters

target

The target to which this command applies. This parameter is mandatory. It must be a single target name, and special characters are not allowed.

-f Force the deletion of the target without requesting confirmation.

Examples

```
nvdm deltg FREDWS
```

This command deletes the target FREDWS from the network.

Related Information

The addtg, lstg, rentg, stattg, and updtg commands.

delusr – Delete a User

Use the **delusr** command to delete a user definition. When a user definition is deleted, all the requests entered by that user are deleted too. If a request entered by that user is pending, a warning message is displayed before you are asked to confirm the deletion.

Syntax

```
▶▶—nvdms delusr—user name—┐ —f ┐▶▶
```

Parameters

user name

The name of the user record to delete.

-f Force the deletion of the user record without requesting confirmation.

The system validates that you are not erasing your own definition or the user root.

Examples

```
nvdms delusr FRED
```

This command deletes the user FRED.

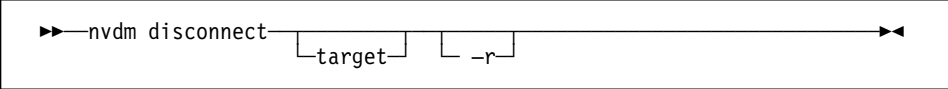
Related Information

The addusr and the updsr commands.

disconnect – Close Mobile Connection with Server

Use the **disconnect** command from a mobile client to halt the connection with a server.

Syntax



```
nvdm disconnect [target] [-r]
```

Parameters

target

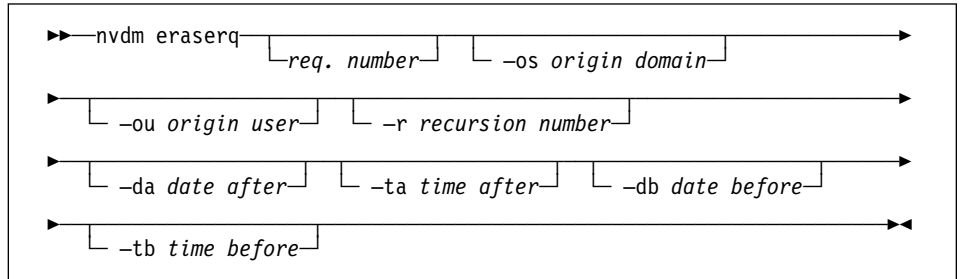
The name of the target you want to disconnect from the server.

- r** Stops the established recursive daily connection from the client to the server that you specified with the **connect target -r** command.

eraserq – Erase Requests

Use the **eraserq** command to remove a request from the database. You can only remove completed requests, whether they were successful or not.

Syntax



Parameters

request number

The sequential number of the request to be removed. You can use an asterisk (*) enclosed in single or double quotation marks to specify all requests.

-os *originator server*

The server where the requests to be removed originated.

-ou *origin user*

The user who submitted the requests. Only the administrator can remove requests submitted by other users. This parameter is optional. The default is the name of the user submitting this command.

-r *recursion number*

Specifies a particular occurrence of a recursive request. If you specify this parameter, you must also specify the **req. num**, that cannot contain the wildcard character. If you do not specify this parameter, only the last occurrence is removed.

-da *date after*

Removes only those requests scheduled after the date specified.

-ta *time after*

Removes only those requests scheduled after the time specified.

-db *date before*

Removes only those requests scheduled before the date specified.

-tb *time before*

Removes only those requests scheduled before the time specified.

Examples

```
nvdn eraserq '*' -ou fred -da 30/06/00
```

This command deletes from the database all those entries submitted by user FRED which were to take place after June 30, 2000.

Related Information

The delrq, lsrq, relrq, and rstrq commands.

exec – Execute a Program or Script

Use the **exec** command to execute, on a target or group, a program or script that is cataloged as a global file. The target must be authorized to use the catalog entry. If you run a script that contains other nvdm commands, insert the following line:

```
nvdm svr <server_name>
```

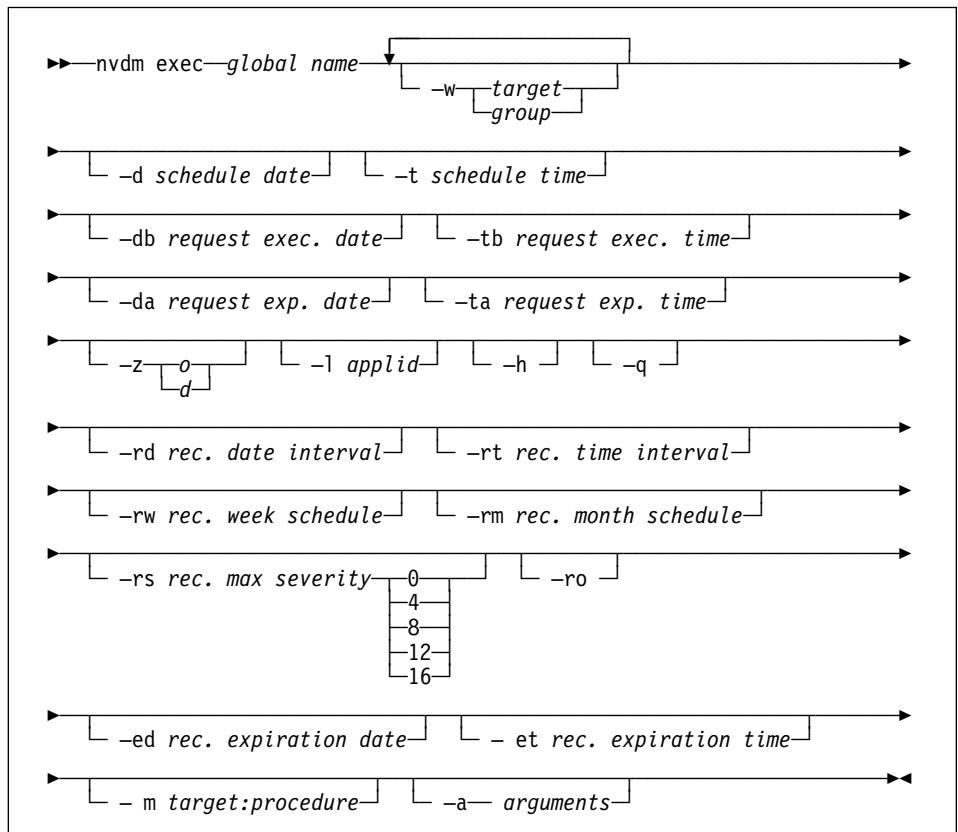
to ensure that the correct connection is made before commands are executed.

For certain types of clients, the types of file that can be executed by this command depend on the type of operating system running on the workstation:

- OS/2 files can have either .CMD or .EXE
- Windows files can have .CMD, .BAT or .EXE

This command cannot be used at mobile clients when working with the local catalog.

Syntax



Parameters

global name

Global name of the program or script to execute. See “Entering Global Names” on page 13. You can use wildcard characters, but a unique match must be found.

-w *target | group*

The targets or groups where the program or script is to be executed. See “Entering Target Names” on page 12.

-d *schedule date*

The date the request is to take place at the origin target.

Date and time parameters are described in more detail in “Entering Dates and Times” on page 13.

-t *schedule time*

The time the request is to take place at the origin target.

-db *request execution date*

The date the request is to take place at the destination target.

-tb *request execution time*

The time the request is to take place at the destination target.

-da *request expiration date*

The date after which the request cannot take place at the destination target.

-ta *request expiration time*

The time after which the request cannot take place at the destination target.

-z *o|d*

The format of the execution time. Enter one of the following values:

- o** The time is expressed as the local time at the origin target. This is the default.
- d** The time is expressed as the local time at the destination target.

-h The request is to be submitted with high priority. You cannot specify this parameter if schedule times and execution times are specified.

-q The request is to be submitted in held state. The request is scheduled only when released by the operator.

This parameter cannot be used locally by mobile clients.

-l *application id*

The ID of the application used to submit the request. This ID can be used later to filter the requests submitted by a specific application.

It does not make sense to set an application ID for a request issued from the graphical interface, as this is an interactive process. Requests submitted from the graphical interface are automatically given the value of GUI as the application ID. You cannot change this value.

If you do not specify an application ID for requests issued from the command-line interface, the default value is CLI.

-rd *rec. date interval*

A recursion date interval expressed in years, months, days. The request is executed upon every lapse of this interval. To specify an interval in terms of days, specify also the months and the years. To specify an interval in terms of months, specify also the years.

For example, 1 means the request is executed every year; 0,1 means the request is executed every month; 0,2,1 means the request is executed every day of every other month.

-rt *rec. time interval*

A recursion time interval expressed in hours and minutes. The request is executed upon every lapse of this interval. The format of a time interval is the same as for a time. For example, 3:45 means a time interval of 3 hours and 45 minutes.

-rw *rec. week schedule*

The days of the week when the recursive request is to take place. The days of the week are as follows:

0 = Sunday
1 = Monday
2 = Tuesday
3 = Wednesday
4 = Thursday
5 = Friday
6 = Saturday

For example, to set a recursion for each Monday and Friday of the week, enter:

-rw 1,5

-rm *rec. month schedule*

The days of the month when the recursive request is to take place. Each day of the month is represented by a number from 1 to 31.

For example, to select a recursion for the 15th and 27th of the month, enter:

-rm 15,27

The number 31 specifies the last day of any month. The number 29 specifies the last day of February even in a non-leap year.

-rs *rec. max severity* **0|4|8|12|16**

A new occurrence of the request is generated only if the severity level returned from the previous occurrence is less than or equal to the max-severity specified.

Specify one of the following:

0 Successful
4 Warning
8 Error
12 Failed
16 Unrecoverable

- ro Specify this parameter to stop the recursion mechanism if the current occurrence of the request is in progress when the new occurrence should be generated. If you do not specify this parameter, one or more occurrences may be skipped until the current one is completed, but the recursion mechanism is not stopped. This parameter is not set by default.
- ed *rec. expiration date*
The date after which the system stops the recursion mechanism.
- et *rec. expiration time*
The time after which the system stops the recursion mechanism.
- m *target:procedure*
A cataloged procedure to execute on a given target when the request is completed. See “Running a Procedure after a Request Has Completed” on page 16.
- a *arguments*
Arguments to be passed to the program or script to be executed. These must be enclosed all together in question marks or double quotation marks.

Examples

```
nvdms exec EURO.VIRUSCHK.EXE.1.US -w FREDSSWS -a “/usr/bin”
```

This command executes the program or script known by the global name EURO.VIRUSCHK.EXE.1.US on FREDSSWS, using the parameter /usr/bin. The procedure is executed as soon as possible.

Related Information

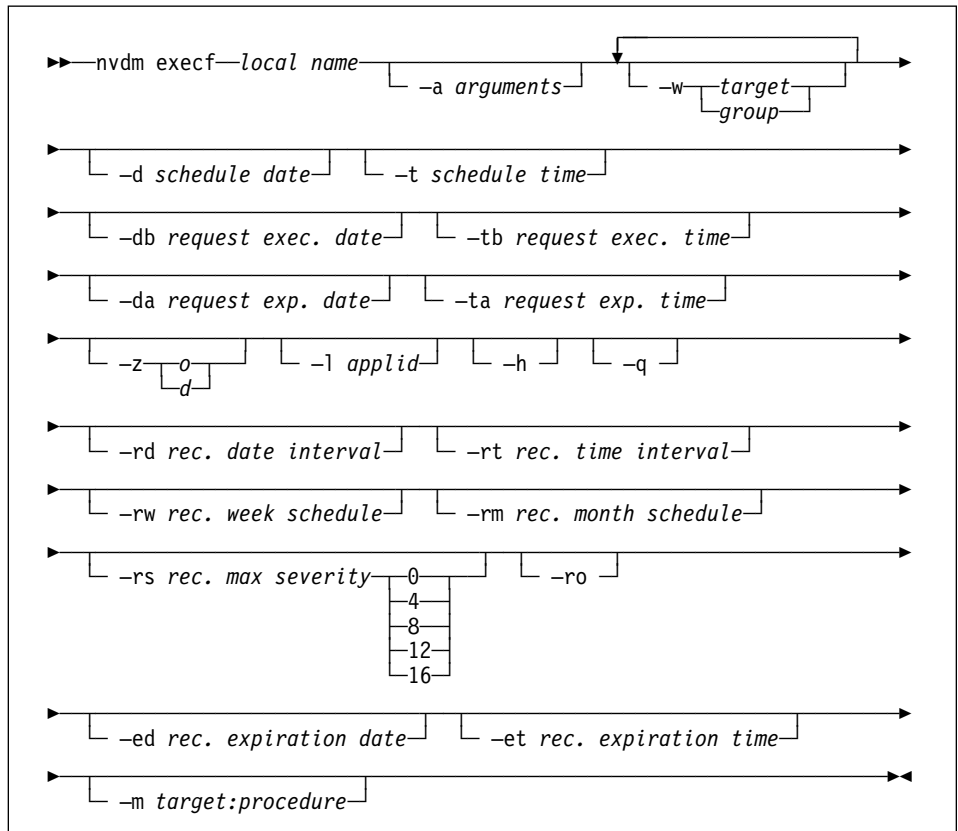
The lsrq command.

execf – Execute a Noncataloged Program or Script

Use the **execf** command to execute a program or script that is not cataloged at any targets. You identify the name of the program or script using its local name.

This command cannot be used at mobile clients when working with the local catalog.

Syntax



Parameters

local name

The local name of the program or script to be executed.

You must specify the file name with its complete path.

Follow the same naming convention that is used at the server to which the destination targets are connected. This parameter is required.

-a arguments

This parameter identifies the arguments to be used by the program or script to execute.

-w *target | group*

The targets or groups where the program or script is to be executed. See “Entering Target Names” on page 12.

-d *schedule date*

The date the request is to take place at the origin target.

Date and time parameters are described in more detail in “Entering Dates and Times” on page 13.

-t *schedule time*

The time the request is to take place at the origin target.

-db *request execution date*

The date the request is to take place at the destination target.

-tb *request execution time*

The time the request is to take place at the destination target.

-da *request expiration date*

The date after which the request cannot take place at the destination target.

-ta *request expiration time*

The time after which the request cannot take place at the destination target.

-z *old*

The format of the execution time. Enter one of the following values:

- o** The time is expressed as the local time at the origin target. This is the default.
- d** The time is expressed as the local time at the destination target.

-h The request is to be submitted with high priority. You cannot specify this parameter if schedule times and execution times are specified.

-q The request is to be submitted in held state. The request is scheduled only when released by the operator.

This parameter cannot be used locally by mobile clients.

-l *application id*

The ID of the application used to submit the request. This ID can be used later to filter the requests submitted by a specific application.

It does not make sense to set an application ID for a request issued from the graphical interface, as this is an interactive process. Requests submitted from the graphical interface are automatically given the value of GUI as the application ID. You cannot change this value.

If you do not specify an application ID for requests issued from the command-line interface, the default value is CLI.

-rd *rec. date interval*

A recursion date interval expressed in years, months, days. The request is executed upon every lapse of this interval. To specify an interval in terms of days,

specify also the months and the years. To specify an interval in terms of months, specify also the years.

For example, 1 means the request is executed every year; 0,1 means the request is executed every month; 0,2,1 means the request is executed every day of every other month.

-rt *rec. time interval*

A recursion time interval expressed in hours and minutes. The request is executed upon every lapse of this interval. The format of a time interval is the same as for a time. For example, 3:45 means a time interval of 3 hours and 45 minutes.

-rw *rec. week schedule*

The days of the month when the recursive request is to take place. The days of the week are as follows:

0 = Sunday
 1 = Monday
 2 = Tuesday
 3 = Wednesday
 4 = Thursday
 5 = Friday
 6 = Saturday

For example, to set a recursion for each Monday and Friday of the week, enter:

-rw 1,5

-rm *rec. month schedule*

The days of the month when the recursive request is to take place. Each day of the month is represented by a number from 1 to 31.

For example, to select a recursion for the 15th and 27th of the month, enter:

-rm 15,27

The number 31 specifies the last day of any month. The number 29 specifies the last day of February even in a non-leap year.

-rs *rec. max severity 0|4|8|12|16*

A new occurrence of the request is generated only if the severity level returned from the previous occurrence is less than or equal to the max-severity specified.

Specify one of the following:

0 Successful
4 Warning
8 Error
12 Failed
16 Unrecoverable

-ro Specify this parameter to stop the recursion mechanism if the current occurrence of the request is in progress when the new occurrence should be generated. If you do not specify this parameter, one or more occurrences may be skipped until the current one is completed, but the recursion mechanism is not stopped. This parameter is not set by default.

-ed *rec. expiration date*

The date after which the system stops the recursion mechanism.

-et *rec. expiration time*

The time after which the system stops the recursion mechanism.

-m *target;procedure*

A cataloged procedure to execute on a given target when the request is completed.

See "Running a Procedure after a Request Has Completed" on page 16.

Examples

```
nvdm execf PGM1.EXE -w target2 -w target3 -w group1 -a "parameter1"
```

This command causes the program PGM1.EXE to be executed at target2, target3, and at all the targets belonging to group1.

```
nvdm execf PGM1.EXE -w target2 -w target3 -w group1 -a "parameter1"
```

This command causes the program PGM1.EXE to be executed at target2, target3, and at all the targets belonging to group1,

```
nvdm execf c:\user\lpp\pgm1.exe -w target2 -w target3▶  
-a "parameter1"
```

This command causes the program PGM1.EXE to be executed at target2 and target3, which are connected to an OS/2 server.

```
nvdm execf c:\usr\lpp\pgm1.exe -w target2 -w target3▶  
-a "parameter1"
```

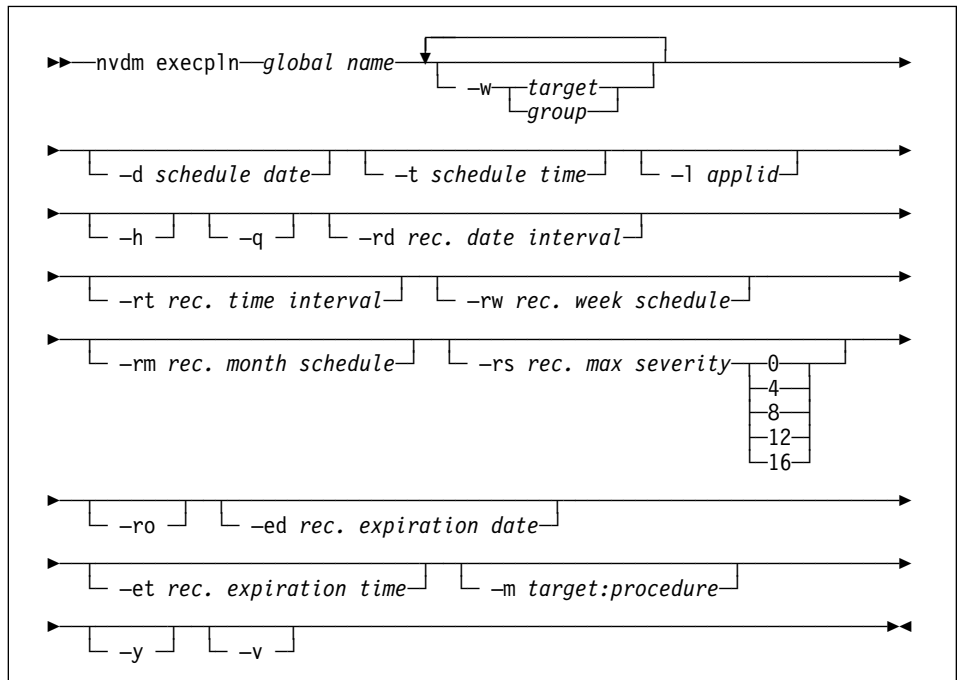
This command causes the program PGM1.EXE to be executed at target2 and target3, which are connected to an NT server.

execpln – Execute a Plan

Use the **execpln** command to execute a plan that is cataloged. The scheduled date and time for the plan execution must not conflict with the execution date and time for any of the entries in the plan.

Note that plan entries that are not conditioned are scheduled for execution in alphanumeric order according to their IDs. Hence, they are not necessarily scheduled in the order they appear in the plan.

Syntax



Parameters

global name

The name of the plan. You cannot use wildcard characters.

-w *target | group*

The targets or groups that are addressed by the plan. See “Entering Target Names” on page 12.

-d *schedule date*

The date the request is to take place at the origin target.

Date and time parameters are described in more detail in “Entering Dates and Times” on page 13.

-t *schedule time*

The time the request is to take place at the origin target.

-h The request is to be submitted with high priority. You cannot specify this parameter if schedule times and execution times are specified.

-q The request is to be submitted in held state. The request is scheduled only when released by the operator.

This parameter cannot be used locally by mobile clients.

-l *application id*

The ID of the application used to submit the request. This ID can be used later to filter the requests submitted by a specific application.

It does not make sense to set an application ID for a request issued from the graphical interface, as this is an interactive process. Requests submitted from the graphical interface are automatically given the value of GUI as the application ID. You cannot change this value.

If you do not specify an application ID for requests issued from the command-line interface, the default value is CLI.

-rd *rec. date interval*

A recursion date interval expressed in years, months, days. The request is executed upon every lapse of this interval. To specify an interval in terms of days, specify also the months and the years. To specify an interval in terms of months, specify also the years.

For example, 1 means the request is executed every year; 0,1 means the request is executed every month; 0,2,1 means the request is executed every day of every other month.

-rt *rec. time interval*

A recursion time interval expressed in hours and minutes. The request is executed upon every lapse of this interval. The format of a time interval is the same as for a time. For example, 3:45 means a time interval of 3 hours and 45 minutes.

-rw *rec. week schedule*

The days of the week when the recursive request is to take place. The days of the week are as follows:

- 0 = Sunday
- 1 = Monday
- 2 = Tuesday
- 3 = Wednesday
- 4 = Thursday
- 5 = Friday
- 6 = Saturday

For example, to set a recursion for each Monday and Friday of the week, enter:

```
-rw 1,5
```

-rm *rec. month schedule*

The days of the month when the recursive request is to take place. Each day of the month is represented by a number from 1 to 31.

For example, to select a recursion for the 15th and 27th of the month, enter:

`-rm 15,27`

The number 31 specifies the last day of any month. The number 29 specifies the last day of February even in a non-leap year.

-rs *rec. max severity* **0|4|8|12|16**

A new occurrence of the request is generated only if the severity level returned from the previous occurrence is less than or equal to the max-severity specified.

Specify one of the following:

- 0** Successful
- 4** Warning
- 8** Error
- 12** Failed
- 16** Unrecoverable

-ro Specify this parameter to stop the recursion mechanism if the current occurrence of the request is in progress when the new occurrence should be generated. If you do not specify this parameter, one or more occurrences may be skipped until the current one is completed, but the recursion mechanism is not stopped. This parameter is not set by default.

-ed *rec. expiration date*

The date after which the system stops the recursion mechanism.

-et *rec. expiration time*

The time after which the system stops the recursion mechanism.

-m *target:procedure*

A cataloged procedure to execute on a given target when the request is completed. See "Running a Procedure after a Request Has Completed" on page 16.

-y Automatically uncatalog the plan after it has been successfully executed.

-v Perform syntax checks to validate the plan without executing it.

Examples

```
nvdn execpln EUR0.PLAN -w group1 -d 20/2/00 -t 8:00 -rw 5 -rs 8
```

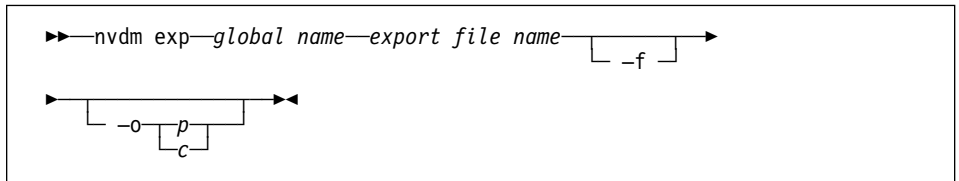
This command causes the plan EUR0.PLAN to be executed at all targets belonging to group1. The first recursion of the plan is executed on February 20, 2000, and will be repeated every Friday from then on. If the severity level returned from a recursion is 8 (ERROR) or higher, the next recursion will not be generated.

exp – Copy a File from the Server to Your Workstation

Use the **exp** command to copy the plan or local file associated with a catalog entry from the server to your workstation and to assign it the name that you specify. The catalog entry is not affected. The **exp** command can be used with plans and change files only. It is used when preparing a plan or a change file for distribution on physical media, such as tape or diskette.

This command cannot be used at mobile clients when working with the local catalog.

Syntax



Parameters

global name

The global name of the change file. You can use wildcard characters, but a unique match must be found. See “Entering Global Names” on page 13.

export file name

The local name of the file to which the change file is copied. If the file already exists, confirmation is requested before overwriting it; otherwise, it is created. The path must already exist.

-f If a local file already exists with the name specified as the *export file name*, the file is overwritten by the new file without a request for confirmation.

-o p|c

The object type. Enter one of the following values:

- p** The object is a plan.
- c** The object is a change file.

Examples

```
nvdm exp EURO.WORDPROC.REF.1.US /u/me/wordproc
```

This command copies the change file EURO.WORDPROC.REF.1.US to file /u/me/wordproc on your computer. The file /u/me/wordproc contains the information about the change file that is needed to catalog it automatically when importing it onto another system.

Related Information

The **imp** command.

help – Display Help for the Command-Line Interface

Use the **help** command to display help information relating to use of the command-line interface.

This command can be used locally by mobile clients.

Syntax



Parameters

command

The name of the command that you want information about. If this is not specified, a list of all the possible commands is displayed.

Examples

```
nvdm help ls
```

This command requests help for the **ls** command. The output is as follows:

```
nvdm ls  global_name  [-o object_type] [-l]
```

hldc – Hold Remote Communication

Use the **hldc** command to hold all remote communications. All distributions to and from the server are held until a **relc** command is issued.

The command can be used to hold communications:

- On SNA/DS connections
- On server-to-server (STS) connections to transfer commands
- On server-to-server (STS) connections to transfer files

Syntax



Parameters

SNADS|CMD|XFER

The communication type to be held. It can be one of the following:

SNADS On SNA/DS connections

CMD On server-to-server (STS) connections to transfer commands

XFER On server-to-server (STS) connections to transfer files

If none of these values is specified, all remote communications are held.

Examples

```
nvdm hldc XFER
```

This command holds all file distributions to and from the server.

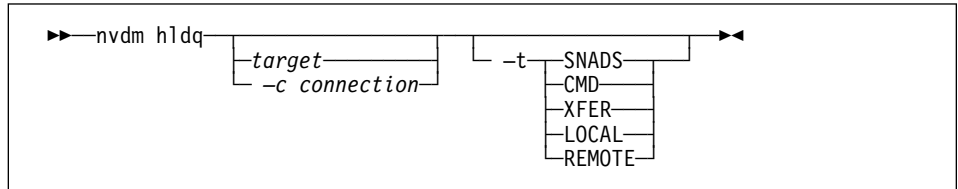
Related Information

The **relc** and **hldq** commands.

hldq – Hold One or All Queues

Use the **hldq** command to hold one or more queues. All distributions on the queues are frozen until a **relq** command is issued.

Syntax



Parameters

target

The name of the target the queue refers to. If the queue is on a local target, this parameter is the queue name; if the queue is on a remote target, this parameter is the name of the next server connected.

If you specify a local target, the **-t** parameter can be omitted. The system recognizes local targets and replaces the correct value for the type of queue.

You can use an asterisk (*) enclosed in single or double quotation marks to hold all the queues of the type specified with the **-t** parameter.

-c connection

The name of the connection the queue refers to. This is the name of the file where the connection is specified, in the `usr/lpp/netviewdm/db/snadscon` directory.

This parameter applies only to queues on remote targets, as local queues are not associated to any connections.

You can use wildcard characters.

-t SNADS|CMD|XFER|LOCAL|REMOTE

The type of queue. It can be one of the following:

- SNADS** On SNA/DS connections
- CMD** On server-to-server (STS) connections to transfer commands
- XFER** On server-to-server (STS) connections to transfer files
- LOCAL** Local
- REMOTE** Remote

If this parameter is not specified, all the remote queues are held. (Their type, in this case, depends on the previous product version installed.)

Examples

```
nvdn hldq new_york
```

This command holds all distributions on the queue called `new_york`.

Related Information

The `relq`, `prgq`, `lsq`, `stat`, and `hldc` commands.

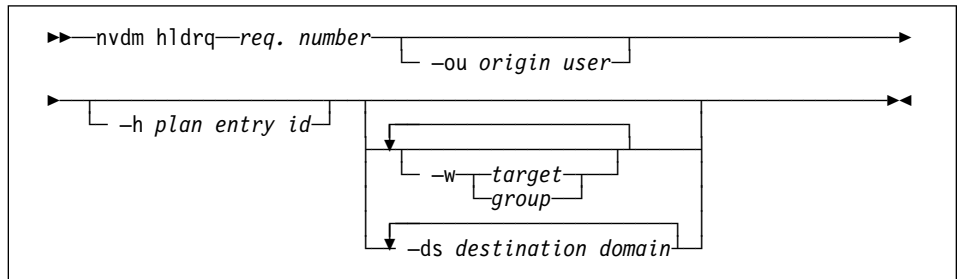
hldrq – Hold a Request

Use the **hldrq** command to hold:

- A request addressed to a specific target or targets
- A request to all targets addressed in a domain
- All requests in a plan

To be held, the request must be in the waiting state.

Syntax



Parameters

request number

The sequential number of the request. This parameter is required.

-ou *origin user*

The user who submitted the request. Only the administrator can hold requests submitted by other users. This parameter is optional. The default is the name of the user submitting this command.

-h *plan entry id*

The identification of the plan entry to be held. If the request does not refer to a plan, an error message is displayed. If you do not specify this parameter and the request refers to a plan, the entire plan is held. If you do not specify either **-w** or **-ds**, all requests for this entry are held.

-w *target | group*

The targets or groups for which the request has to be held. See “Entering Target Names” on page 12.

This parameter cannot be specified if **-ds** is specified.

-ds *destination domain*

The domain for which the request has to be held. This parameter cannot be specified if **-w** is specified.

Examples

```
nvdm hldrq 13 -w FREDWS
```

This command holds the request whose number is 13 for target FREDWS.

Related Information

The delrq, eraserq, relrq, and rstrq commands.

hldtg

hldtg – Hold a Target

Use the **hldtg** command to hold a local target. When a target is held, only the user of the target itself can perform change control operations on it. Other users can schedule operations for the held target but only distribution requests are carried out; change control requests are not performed until the target is released. Pull-mode targets cannot be held.

If the system is halted and then restarted, a target retains the held status.

Syntax



Parameters

target | group

Target or group to be held. This parameter is optional. You can use an asterisk (*) enclosed in single or double quotation marks to hold all targets.

Examples

```
nvdm hldtg FREDSSWS
```

This command holds the target named FREDSSWS.

Related Information

The `reltg` command.

imp – Import a Change File into the System

Use the **imp** command to import a noncataloged change file or plan from your computer to the system server and catalog it. The file is cataloged using information that is stored within the change file. Any information concerning the products and filesets contained in the change file is cataloged as well. Use this command when introducing a change file that was distributed on physical media, such as tape or diskette, into the system.

This command can be used locally by mobile clients. An additional parameter, **-n**, can be used only with mobile clients to specify that a catalog entry should be created but the change file should remain on the external device. A report of the completed operation is forwarded to the mobile client's server.

Syntax

```

▶▶—nvdm imp—import file name— [ -f ] [ -n ] [ -o p|c ] ▶▶

```

Parameters

import file name

The local name of the change file that is to be imported. This is the name of the file on your computer, not the local name of the file after it has been imported.

If the catalog entry already exists, the command is unsuccessful.

- f** Forces an overwrite if the file already exists. If **-f** is not specified, you will be asked to confirm the overwrite.
- n** This parameter can only be used locally at mobile clients. It specifies that a catalog entry should be created for the change file, using the information contained in the change file, but the change file itself should remain on the external device.
- o p|c**
The object type. Enter one of the following values:
 - p** The object is a plan.
 - c** The object is a change file.

Examples

```
nvdm imp /u/me/amword
```

This command catalogs the change file held in `/u/me/amword` and copies it to the server.

imp

Related Information

The `exp` and `bld` commands.

inst – Install a Change File

Use the **inst** command to request the installation of up to seven change files. The change files you specify are treated as corequisites, which means that either all installations succeed or all do not. Options on the command specify:

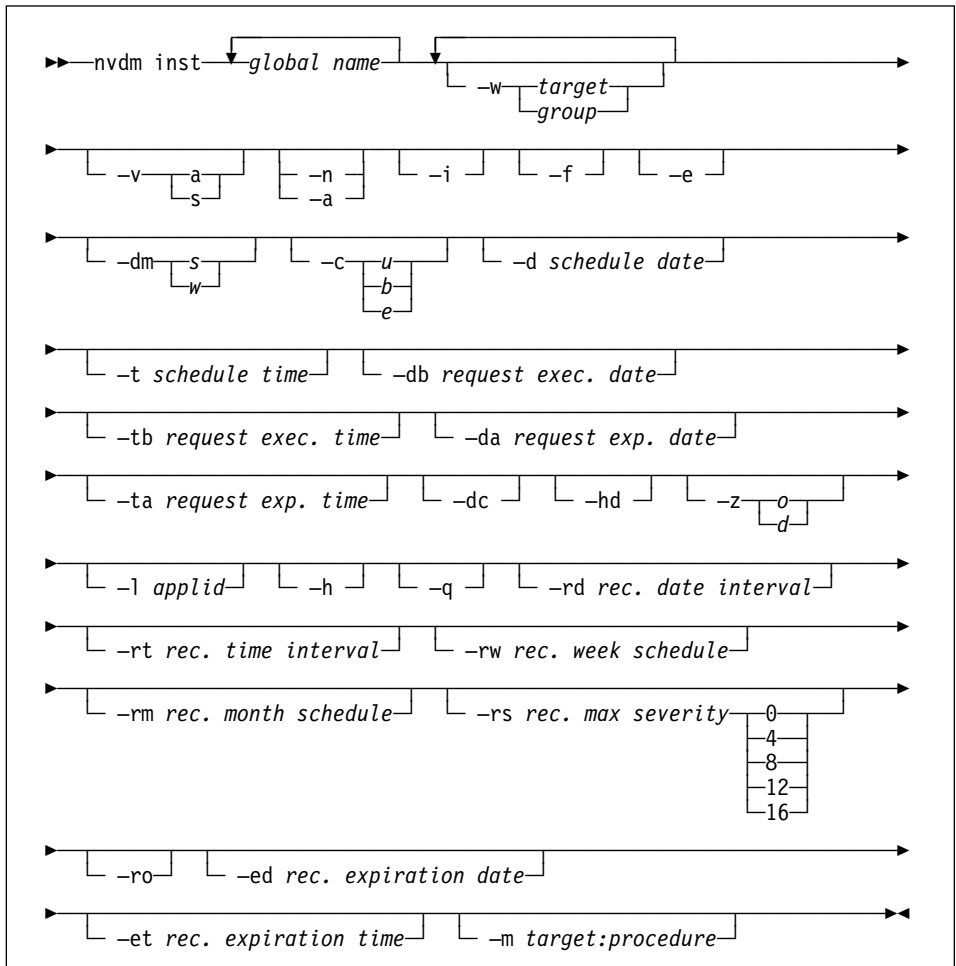
- The target or group on which the installation is to be performed.
- Whether the installation will be completed by an activation, in which case the installation is done to the service area. If an activation is not required, the installation should be done direct to the active area.
- Whether the change file installation is:
 - Removable
 - Automatically accepted
- The date and time at which the installation takes place.
- Whether the installation procedure should proceed when identical files are found at previous or at higher release levels.

If you are authorized to use files, the change file does not need to be authorized for use by the target (this will be performed automatically).

Files to be installed can be stored in a remote directory at the server, if the targets you are installing on have been configured with specific tokens that mount the directory before performing the installation. Use the **addpm** command to define these tokens. When remote directories are mounted for an installation, no disk space is required on the target to store files before they are actually installed.

This command can be used locally by mobile clients. It does not, however, accept the **-w**, **-h**, **-z**, **-rd**, **-rt**, **-rw**, **-rm**, **-rs**, **-ro**, **-ed**, **-et** or **-m** parameters.

Syntax



Parameters

global name

Global names of the change files to install. You can specify up to seven change file names; they are installed as corequisites in the order in which they are specified.

You can use wildcard characters, but the installation is scheduled only if a unique match is found for each file name.

-w *target | group*

The targets or groups to which this command applies. See “Entering Target Names” on page 12.

This parameter cannot be used locally by mobile clients.

-v a|s

Disk area where the change file is to be installed. Possible values are:

a Install to active area.

s Install to service area. An activation is required to complete the installation.

If **-v** is not specified, the installation is performed to the active area.

-n Install the change file nonremovably, making no backups of any files that are changed by the installation process.

If **-n** is not specified, the change file is installed removably.

-a Automatically accept the installation when it is complete. This causes all backup files to be deleted. **-a** is valid only when **-n** is not specified.

If **-a** is not specified, the installation is not automatically accepted.

-i Force the Install command to be scheduled even if the change file history on the target does not allow it. However, the installation will not succeed if the change file history forbids it at the time when the installation is to be executed.

-f Force the installation to take place even if the change file history forbids it. This is similar to the **-i** option, except that the check of the change file history at the time of installation is bypassed.

-e Extend file systems. If this parameter is specified, the journaled file system at the target is extended during the installation of the change file if the installation operation detects that additional disk space is required.

-dm driver mode s|w

This parameter determines one of the following installation behaviors:

s Stop the installation if the condition specified by the **-c** parameter is true. If this is the case, the installation fails with return code 12.

w If the condition specified by the **-c** parameter is true, do not stop the installation, but issue a warning message and *do not* replace the object that caused the error. If this is the case, the installation is completed with return code 4.

For both parameters, a message is logged in `fnldlog` that reports the object that caused the error. You can issue the **lsrq** command with the **-w** parameter to display the error message and the first object encountered that caused it.

-c condition u|b|c

The condition that must be checked for each object in the change file, and which determines whether the installation is to be stopped or to proceed, according to the values specified for the **-dm** parameter. Note that the condition is checked on change files present in both the service area and the work area at a target (see the **-v** parameter).

If **-dm** is not specified, the installation will proceed even if the condition specified is verified for some objects in the change file. If this is the case, the objects are not replaced, and the request is completed as successful.

Specify one of the following values:

- u** *install if file is uplevel* If the file being installed does not exist on the target, install it. If the file does exist, check whether its date is older or newer than the file being installed. If it is newer, install the file. If it is older, do not install the file.
- b** *install if file is backlevel* If the file being installed does not exist on the target, install it. If the file does exist, check whether its date is older or newer than the file being installed. If it is older, install the file. If it is newer, do not install the file.
- e** *install if file does not exist already* If the file being installed does not already exist on the target, install it. If the file does exist, do not install it.

-d *schedule date*

The date the request is to take place at the origin target.

Date and time parameters are described in more detail in “Entering Dates and Times” on page 13.

-t *schedule time*

The time the request is to take place at the origin target.

-db *request execution date*

The date the request is to take place at the destination target.

-tb *request execution time*

The time the request is to take place at the destination target.

-da *request expiration date*

The date after which the request cannot take place at the destination target.

-ta *request expiration time*

The time after which the request cannot take place at the destination target.

-dc

This parameter refers to operations performed at mobile clients. When it is specified, the operation is to be performed when the mobile client is disconnected from the server.

-hd

This parameter pertains to operations performed at mobile clients, and can only be specified together with the **-dc** parameter. It holds the disconnected request at the server until its execution time is reached. The request is forwarded to the client during the next connection window opened after the execution time specified, and executed there when the client is disconnected.

-z o|d

The format of the execution time. Enter one of the following values:

- o** The time is expressed as the local time at the origin target. This is the default.
- d** The time is expressed as the local time at the destination target.

This parameter cannot be used locally by mobile clients.

- h The request is to be submitted with high priority. You cannot specify this parameter if schedule times and execution times are specified.

This parameter cannot be used locally by mobile clients.

- q The request is to be submitted in held state. The request is scheduled only when released by the operator.

This parameter cannot be used locally by mobile clients.

-l *application id*

The ID of the application used to submit the request. This ID can be used later to filter the requests submitted by a specific application.

It does not make sense to set an application ID for a request issued from the graphical interface, as this is an interactive process. Requests submitted from the graphical interface are automatically given the value of GUI as the application ID. You cannot change this value.

If you do not specify an application ID for requests issued from the command-line interface, the default value is CLI.

-rd *rec. date interval*

A recursion date interval expressed in years, months, days. The request is executed upon every lapse of this interval. To specify an interval in terms of days, specify the months and the years. To specify an interval in terms of months, specify the years.

For example, 1 means the request is executed every year; 0,1 means the request is executed every month; 0,2,1 means the request is executed every day of every other month.

This parameter cannot be used locally by mobile clients.

-rt *rec. time interval*

A recursion time interval expressed in hours and minutes. The request is executed upon every lapse of this interval. The format of a time interval is the same as for a time. For example, 3:45 means a time interval of 3 hours and 45 minutes.

This parameter cannot be used locally by mobile clients.

-rw *rec. week schedule*

The days of the week when the recursive request is to take place. The days of the week are as follows:

- 0 = Sunday
- 1 = Monday
- 2 = Tuesday
- 3 = Wednesday
- 4 = Thursday
- 5 = Friday
- 6 = Saturday

For example, to set a recursion for each Monday and Friday of the week, enter:

```
-rw 1,5
```

This parameter cannot be used locally by mobile clients.

-rm *rec. month schedule*

The days of the month when the recursive request is to take place. Each day of the month is represented by a number from 1 to 31.

For example, to select a recursion for the 15th and 27th of the month, enter:

```
-rm 15,27
```

The number 31 specifies the last day of any month. The number 29 specifies the last day of February even in a non-leap year.

This parameter cannot be used locally by mobile clients.

-rs *rec. max severity* **0|4|8|12|16**

A new occurrence of the request is generated only if the severity level returned from the previous occurrence is less than or equal to the max-severity specified.

Specify one of the following:

- 0** Successful
- 4** Warning
- 8** Error
- 12** Failed
- 16** Unrecoverable

This parameter cannot be used locally by mobile clients.

- ro** Specify this parameter to stop the recursion mechanism if the current occurrence of the request is in progress when the new occurrence should be generated. If you do not specify this parameter, one or more occurrences may be skipped until the current one is completed, but the recursion mechanism is not stopped. This parameter is not set by default.

This parameter cannot be used locally by mobile clients.

-ed *rec. expiration date*

The date after which the system stops the recursion mechanism.

This parameter cannot be used locally by mobile clients.

-et *rec. expiration time*

The time after which the system stops the recursion mechanism.

This parameter cannot be used locally by mobile clients.

-m *target;procedure*

A cataloged procedure to execute on a given target when the request is completed. See "Running a Procedure after a Request Has Completed" on page 16.

This parameter cannot be used locally by mobile clients.

Examples

```
nvdn inst EURO.WORDPROC.REF.2.US EURO.WORDPROC.UPD.2.3.US -w  
FREDSWS -va -d "2/4/00" -t "03:00"
```

This command installs the two change files EURO.WORDPROC.REF.2.US and EURO.WORDPROC.UPD.2.3.US on target FREDSWS with the following attributes:

- The installation is performed in the active area
- The installation is removable
- The installation is not automatically accepted
- Journaled file systems will not be extended
- The installation is scheduled for 3 a.m. on 2 April 2000

Related Information

The lsrq, acc, rem, and uninst commands.

inv – Inventory Discovery

Use the **inv** command to perform inventory discovery on a target or a group of targets. Inventory discovery information (including information on the product and fileset) is stored in the `fnswinv`, `fnhwinv`, and `fnltkinv` files at the target where the inventory is performed.

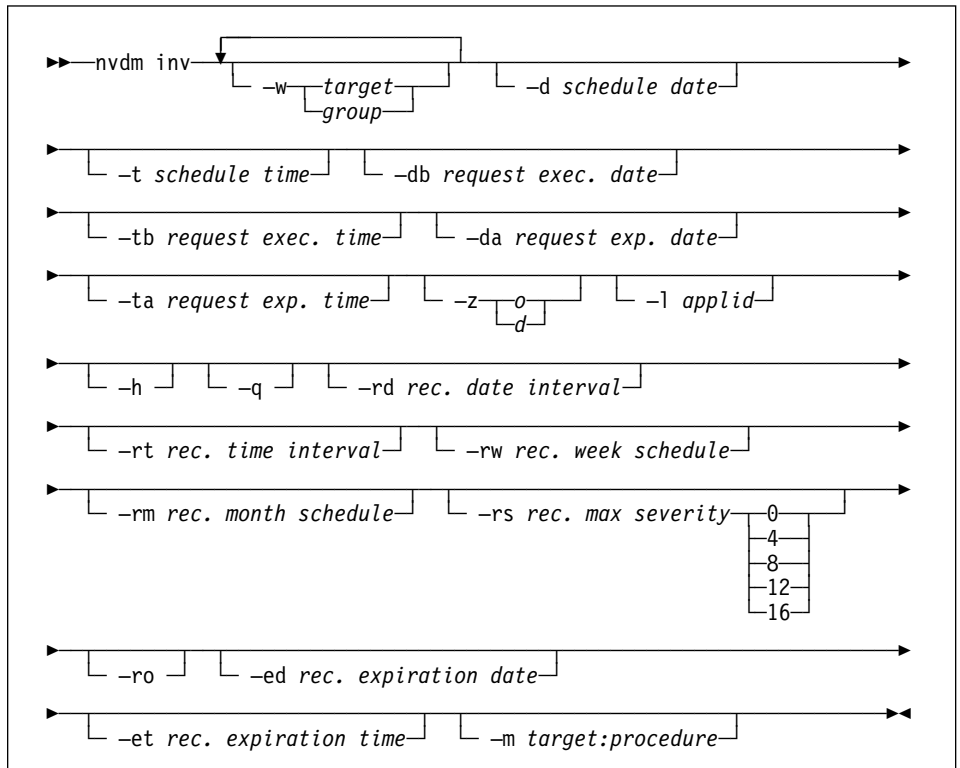
This operation can be performed on remote targets only if the targets involved in the operation are connected by the server-to-server (STS) transmission protocol. This procedure is normally executed by a script after running the inventory discovery program. You need Modify Configuration authorization to issue this command.

Hardware inventory information is stored as part of the target configuration record. Software inventory data results in the creation of a catalog entry and status record with the status of discovered, meaning that the software is installed, but was not installed by TME 10 Software Distribution. Discovered software is active and not uninstallable.

The discovered hardware inventory is completely refreshed each time this command is run. The discovered software inventory is cumulative. That is, entries can only be added using this command. To delete a discovered software package from the inventory, you must remove it from the catalog.

Inv is scheduled automatically when a new target is added using either the graphical user interface or the **addtg** command.

Syntax



Parameters

-w *target | group*

The targets or groups where the inventory discovery is to be performed. See “Entering Target Names” on page 12.

-d *schedule date*

The date the request is to take place at the origin target.

Date and time parameters are described in more detail in “Entering Dates and Times” on page 13.

-t *schedule time*

The time the request is to take place at the origin target.

-db *request execution date*

The date the request is to take place at the destination target.

-tb *request execution time*

The time the request is to take place at the destination target.

-da *request expiration date*

The date after which the request cannot take place at the destination target.

-ta *request expiration time*

The time after which the request cannot take place at the destination target.

-z o|d

The format of the execution time. Enter one of the following values:

- o** The time is expressed as the local time at the origin target. This is the default.
- d** The time is expressed as the local time at the destination target.

-h The request is to be submitted with high priority. You cannot specify this parameter if schedule times and execution times are specified.**-q** The request is to be submitted in held state. The request is scheduled only when released by the operator.

This parameter cannot be used locally by mobile clients.

-l *application id*

The ID of the application used to submit the request. This ID can be used later to filter the requests submitted by a specific application.

It does not make sense to set an application ID for a request issued from the graphical interface, as this is an interactive process. Requests submitted from the graphical interface are automatically given the value of GUI as the application ID. You cannot change this value.

If you do not specify an application ID for requests issued from the command-line interface, the default value is CLI.

-rd *rec. date interval*

A recursion date interval expressed in years, months, days. The request is executed upon every lapse of this interval. To specify an interval in terms of days, specify also the months. To specify an interval in terms of months, specify also the years.

For example, 1 means the request is executed every year; 0,1 means the request is executed every month; 0,2,1 means the request is executed every day of every other month.

-rt *rec. time interval*

A recursion time interval expressed in hours and minutes. The request is executed upon every lapse of this interval. The format of a time interval is the same as for a time. For example, 3:45 means a time interval of 3 hours and 45 minutes.

-rw *rec. week schedule*

The days of the week when the recursive request is to take place. The days of the week are as follows:

0 = Sunday
 1 = Monday
 2 = Tuesday
 3 = Wednesday
 4 = Thursday
 5 = Friday
 6 = Saturday

For example, to set a recursion for each Monday and Friday of the week, enter:

`-rw 1,5`

-rm *rec. month schedule*

The days of the month when the recursive request is to take place. Each day of the month is represented by a number from 1 to 31.

For example, to select a recursion for the 15th and 27th of the month, enter:

`-rm 15,27`

The number 31 specifies the last day of any month. The number 29 specifies the last day of February even in a non-leap year.

-rs *rec. max severity* **0|4|8|12|16**

A new occurrence of the request is generated only if the severity level returned from the previous occurrence is less than or equal to the max-severity specified.

Specify one of the following:

0 Successful
4 Warning
8 Error
12 Failed
16 Unrecoverable

-ro Specify this parameter to stop the recursion mechanism if the current occurrence of the request is in progress when the new occurrence should be generated. If you do not specify this parameter, one or more occurrences may be skipped until the current one is completed, but the recursion mechanism is not stopped. This parameter is not set by default.

-ed *rec. expiration date*

The date after which the system stops the recursion mechanism.

-et *rec. expiration time*

The time after which the system stops the recursion mechanism.

-m *target:procedure*

A cataloged procedure to execute on a given target when the request is completed. See “Running a Procedure after a Request Has Completed” on page 16.

Examples

```
nvdm inv -w target1
```

This command provides the following output.

```
GLOBAL NAME:      NETWORK.PLAN.REF.3
DESCRIPTION:      Network planner 3.1
  ARCHITECTURE:    Platform architecture
  VENDOR TAG:      IBM
  TITLE:           Network Planner
  FILESET:
    TAG:           Base
    REVISION:      3.1.1
    TITLE:         Base feature
  FILESET:
    TAG:           Samples
    REVISION:      3.1.5
    TITLE:         Sample feature

GLOBAL NAME:      NETWORK.ADMN.REF.2
DESCRIPTION:      Network Administrator 2.1
CHANGE FILE TYPE: GEN

PRODUCT:
  TAG:            NETADMIN
  REVISION:       2.1
  ARCHITECTURE:    Platform architecture
  VENDOR TAG:      IBM
  TITLE:           Network Administrator
  FILESET:
    TAG:           Administration
    REVISION:      2.1.1
    TITLE:         Administration feature
  FILESET:
    TAG:           Samples
    REVISION:      2.1.4
    TITLE:         Sample feature
```


log – Display the Contents of the Message Log

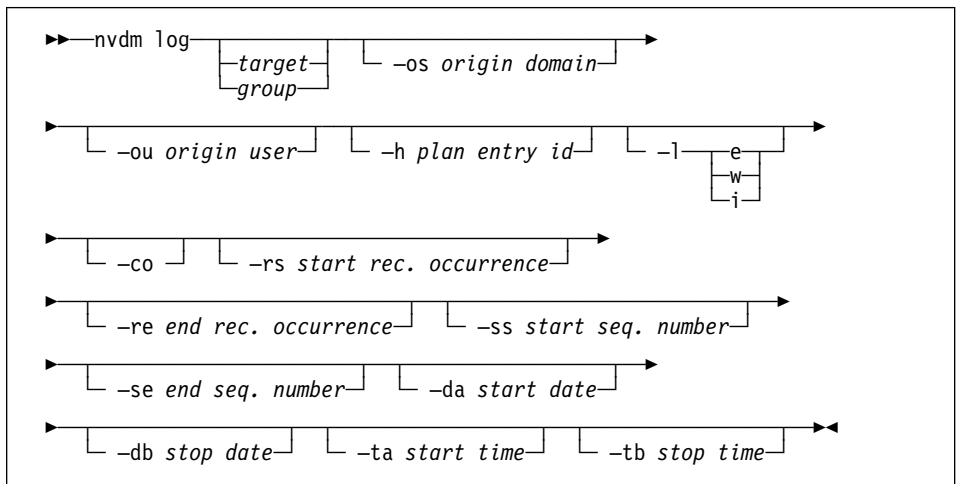
Use the **log** command to display messages contained in the message log. You can use parameters to limit the messages displayed to ones that meet certain criteria.

The messages are listed in reverse chronological order, with the most recent message displayed first.

Parameters **-os**, **-ou**, **-h**, **-rs**, **-re**, **-ss**, **-se** apply only to messages having correlators.

This command cannot be used at mobile clients when working with the local catalog.

Syntax



Parameters

target | group

Limits the display to messages associated with the specified target or group. Messages are displayed only if:

- The target or group was the source of the message, or
- The message includes a correlator, and:
 - The target or group was the issuer of the request
 - The target or group was the destination of the request

You can use an asterisk (*) enclosed in single or double quotation marks to display messages associated with all targets.

The default is the target where this command is submitted.

-os *origin domain*

Limits the display to requests submitted from the specified server's domain. You can use an asterisk (*) enclosed in single or double quotation marks to indicate all domains.

The default is the server connected to the target where this command is submitted.

-ou *origin user*

Limits the display to messages related to requests submitted by the specified user. You can use an asterisk (*) enclosed in single or double quotation marks to indicate all users.

The default is the name of the user submitting this command.

-h *plan entry id*

Limits the display to messages associated with the specified plan entry. You can use an asterisk (*) enclosed in single or double quotation marks to indicate all plan entries.

The default is all plan entries.

-l e|w|i

The level of messages to display:

- e** Error messages only
- w** Warning and error messages
- i** Information, warning, and error messages

The default value is **w**.

-co

Only messages including correlators are displayed.

-rs *start recursion occurrence*

The occurrence number of a recursive request starting from which you want to display messages. If the same value is set for the **-re** parameter, messages related only to that occurrence are displayed.

The default value is 0, meaning all requests (including those that are not recursive).

-re *end recursion occurrence*

The occurrence number of a recursive request up to which you want to display messages. If the same value is set for the **-rs** parameter, messages related only to that occurrence are displayed.

The default is 65 535.

-ss *start sequence number*

The number starting from which you want to display messages. If the same value is set for the **-se** parameter, messages related only to that sequence number are displayed.

The default value is 1, meaning the first request that was submitted.

-se *end sequence number*

The number up to which you want to display messages. If the same value is set for the **-ss** parameter, messages related only to that sequence number are displayed.

The default value is 4 294 967 295.

-da *start date*

The date starting from which the messages issued are to be displayed. The list is in reverse chronological order, so these messages are found at the end of the list. The default is the current date.

See “Entering Dates and Times” on page 13.

-db *stop date*

The date up to which the messages issued are to be displayed. The default is the end of log.

-ta *start time*

The time starting from which the messages issued are to be displayed. The default is the current time.

-tb *stop time*

The time up to which the messages issued are to be displayed. The default is the end of log.

Examples

```
nvdm log -w FREDWS -lw
```

This command shows all messages issued from FREDWS that were error or warning messages, starting with the most recent one.

Related Information

The addpm command.

ls – List Entries in the Catalog

Use the **ls** command to list entries in the catalog. All catalog entries that match the specified global name, and that you are authorized to use, are displayed. Information about the products and filesets contained in the entry is also shown.

When you find a catalog entry that has the following information:

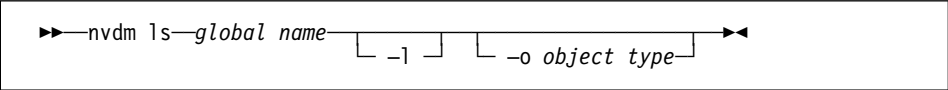
Object at server: not committed

the file has not completely arrived at the server during a build or send process.

In this case you cannot perform an install, initiate, send, or retrieve request on it.

This command can be used locally by mobile clients.

Syntax



Parameters

global name

The global name of the entries to be listed. See “Entering Global Names” on page 13. You can use wildcard characters.

-l Provides a longer format display, including the compression parameters.

-o *object type*

The type of object you are requesting the list for. The possible values, with allowed abbreviations, are in the following table. You can also represent the object type using its SNA/FS class code. Specify 0x followed by the corresponding 8 hexadecimal characters in C language format.

Table 7. Object Types and their SNA/FS Class Codes	
Object Type	Hexadecimal value
FLATDATA (FLATD)	0x20100000
SOFTWARE (SOFTW)	0x10300000
MICROCODE (MICR)	0x10100000
PROCEDURE (PROC)	0x10500000
RELDATA (RELD)	0x20200000
DUMP	0x40100000
CONFIGFILE (CONFIG)	0x40200000
TRACE	0x40300000
ERRLOG	0x40400000
PLAN	0x00801050

The default object type is FLATDATA.

Output

The data for each matching component name is output as follows.

Short Form

```
Description:      [description]
Global Name:      [global name]
Local Name:       [local name]
Object Type:      [object type]
```

Additional on Long Form

```
Compression:      [compression type]
Currently compressed: [Yes|No]
```

Examples

```
nvdms ls EURO.WORDPROC.* -l
```

This command lists all catalog entries beginning EURO.WORDPROC.*, displaying output in the long form. The output looks like the following:

```
Description:      Word processor level 1.1
Global Name:      EURO.WORDPROC.REF.1.US
Local Name:       /usr/lpp/nvdm/repos/EURO.WORDPROC.REF.1.US
Additional local name: (none)
Object Type:      SOFTWARE
Data access key:  (none)
Compression:      LZW
Currently compressed: Yes
Product:          (none)

Description:      Word processor level 2
Global Name:      EURO.WORDPROC.REF.2.US
Local Name:       /usr/lpp/nvdm/repos/EURO.WORDPROC.REF.2.US
Additional local name: (none)
Object Type:      SOFTWARE
Data access key:  (none)
Compression:      SNA
Currently compressed: No
Product:          (none)
```

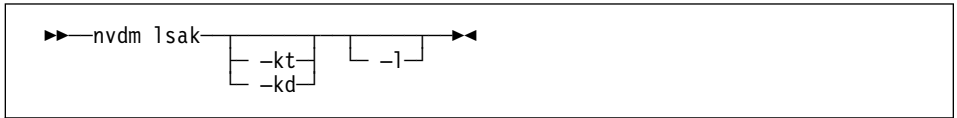
Related Information

The cat, imp, and bld commands.

lsak – List Access Keys

Use the **lsak** command to list the names and descriptions of the data access keys or target access keys defined.

Syntax



Parameters

- kt** *target access key*
List target access keys.
- kd** *data access key*
List data access keys.
- l** Long mode. If you specify this parameter, the complete description for the access key is listed.

Examples

```
nvdm lsak -kt -l
```

This command produces the following output:

Target Access Key:	TAK01
Description:	Access key for personnel.
Target Access Key:	TAK02
Description:	Access key for development.
Target Access Key:	TAK02
Description:	Access key for administration.

Related Information

The updak command.

lsbs – List Configuration

Use the **lsbs** command to report configuration information for the base option and the server option. It shows whether you have the base option, the server option, or the remote administration option installed, whether SNA communication is available, and whether authorization of TCP/IP addresses is required. In addition, it reports the name of the server that you are currently logged on.

You must have the View or Modify System Administration authorization or the View or Modify System Configuration to use this command.

Syntax

```
nvdm lsbs
```

Output

The output of the **lsbs** command is as follows.

```
Configuration:          <Base/Server>
Base|Server name:      [name]
Remote administration:  <Yes/No>
Remote communications:  <Yes/No>
LAN authorization:      <Yes/No>
Authorize:              <All|NONE>
```

Either the base name or the server name is provided as output, depending on the configuration.

Examples

```
nvdm lsbs
```

This command lists the base or server configuration. The output looks like the following.

```
Configuration:          Server
Server name:            back_office
Remote administration:   No
Remote communications:   No
LAN authorization:       Yes
Authorize:               NONE
```

Related Information

The updb command.

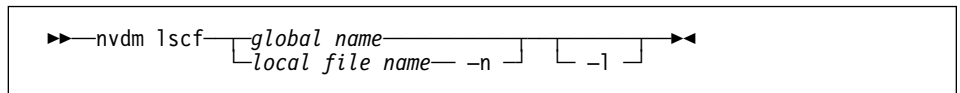
lscf – List the Contents of a Change File

Use the **lscf** command to list the contents of a change file. The report can be produced in the same format as a change file profile. This means that it performs exactly the opposite function from the **bld** command.

The contents of the change file are written to standard output in the same format as defined in the change file profile.

This command cannot be used at mobile clients when working with the local catalog.

Syntax



Parameters

global name

The global name of the change file to be reported. You can use wildcard characters, but a unique match must be found.

local file name -n

The name of a change file on your local workstation. Specify the **-n** parameter to differentiate this name from the global name.

- l** Expand any wildcard object definitions to show each of the files that were included in the change file as a result of the wildcard. If you specify this option, the output is no longer in the correct format for a change file profile.

This option also produces a long listing of change file contents that includes the attributes for files that were copied and directories that were created.

The following information is displayed, even if it is not in the original change file profile:

Creation Date The change file creation date and time. It is displayed in the format YY/MM/DD HH:MM:SS.

If the change file was migrated from a previous release, the creation date is not available.

Layout The versions of the product on which the change file can be used. The following values can apply:

- 1 Version 1.0, or higher
- 2 Version 1.1, or higher
- 3 Version 3.1, or higher
- 4 Version 3.1.5

Examples

```
nvdms lscf EURO.WORDPROC.REF.1.US
```

This command reports the contents of the change file EURO.WORDPROC.REF.1.US. The output is produced in the same format as defined in the change file profile.

```
nvdms lscf NETWORK.PLAN.REF.3 -1
```

This command produces a long listing that includes the following information:

```
CREATION DATE:      00/07/24   12:21:00
LAYOUT:             3
GLOBAL NAME:        NETWORK.PLAN.REF.3
LOCAL NAME:         $(REPOSITORY)/NETWORK.PLAN.REF.3
```

```
CHANGE FILE TYPE:   GEN
COMPRESSION TYPE:   LZW
REBOOT REQUIRED:     YES
REMOVABLE:          NO
ACTIVATABLE:        YES
INTERACTIVE:        YES
AUTHORIZE:          ALL
INSTALLATION DURATION: 4
SW HISTORY RESET:   YES
COST:               1500
PACK FILES:         NO
SECURE PACKAGE:     NO
```

OBJECT:

```
SOURCE NAME:       /usr/lpp/netviewdm/fndnoss
TARGET NAME:        /usr/lpp/netviewdm/fndnoss
TYPE:               FILE
ACTION:             COPY
INCLUDE SUBDIRS:    NO
OWNER:              root
GROUP:              FNDADMN
GENERAL ATTRIBUTES: -----
UNIX ATTRIBUTES:    -rw-rw-----
NETWARE ATTRIBUTES: -----NDC
SIZE (bytes):       272
```

No information is held in the change file for objects that are being deleted; therefore, no information is shown in the example above.

The only information shown for directories that are being created is owner, group, and mode. Date and time are not shown because the directories are created with the current date and time when the change file is created. Size is not relevant for directories.

For remote files, attribute information is only shown if the change file was built with the Secure Package option set to yes. If not, no attribute information about the file is held in the change file.

lscf

Related Information

The bld command.

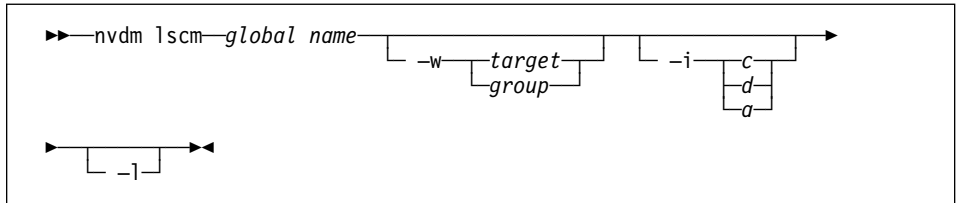
Chapter 3, "Creating Change File Profiles" on page 247.

lscm – Report the Status of Files

Use the **lscm** command to request a status report for one or more files on a target or group of targets.

This command can be used locally by mobile clients. However, the **-w** parameter cannot be used.

Syntax



Parameters

global name

Global name of the file for which the status is to be displayed. See “Entering Global Names” on page 13. You can use wildcard characters.

-w *target | group*

The targets or groups to which this command applies.

You cannot specify more than one target or group at a time.

You can use an asterisk (*) enclosed in single or double quotation marks to show the status of the file on all targets. See “Entering Target Names” on page 12.

This parameter cannot be used locally by mobile clients.

-i *c|d|a*

Include file types.

- c** Change files
- d** Data files
- a** All files

If **-i** is not specified, only change files are shown.

-l List additional information such as:

- The name of the user who issued the last request
- The date and time of the last request
- The last change control request started, and its status

You cannot use this option **-id** is specified because the information is related to change files only.

Output

The output of the **lscm** request is as follows.

The following header line is output for each change file matching the supplied global name.

Global name: [global name]

The following block of lines is output for each target requested.

Target: [target]

Status: [status see below]

This additional information is provided if **-l** is specified:

Last request: [Install/Accept/Remove/Uninstall/Activate] [Executing]
[Back-leveled] [Restored] [Activated by reboot] [Superseded]
[Discovered]
Issued by: [User|Unknown] [|at Focal point|at Remote]
Execution date: [date/time]

The status can be one or a combination of the following:

OVERALL STATUS

- OK** The change management action on the change file completed successfully.
- In error** The change file underwent a change management action that failed without recovering, leaving the installation in an unpredictable state. In this case, you can attempt all change management operations.
- Not authorized** The change file is cataloged on the server but the target in question is not authorized to use it.

INSTALL STATUS

- Available** The change file is cataloged on the server and the target in question is authorized to use it.
- Back level** The change file is a previous version of the software that was backed up during the installation of a more recent version.
- Discovered** The change file was discovered by an inventory discovery procedure. The change file was installed, but not by TME 10 Software Distribution.
- Distributed** The change file has been distributed to this target from another target.
- Distribution pending** The change file is currently being distributed.
- Scheduled** The installation of the change file has started but not yet completed.
- Installed** The change file is installed.

REMOVABILITY STATUS

Removable The change file is installed in a manner that allows it to be removed later. True if the change file was installed removably on the target and not subsequently uninstalled or removed.

Not removable
The change file is installed and is not removable.

ACTIVE STATUS

Active The change file is installed in the active area of the disk.

Inactive The change file is installed in the service area awaiting activation.

Reboot required
Indicates that a change file has been installed that modified a file which is read only when the workstation is restarted. To use the new software, you must restart the workstation.

Data files are shown if they have an Available or Distributed status. The other states are not relevant for data files.

Examples

```
nvdms lscm EURO.WORDPROC.REF.* -w*
```

This command reports the status of the change file EURO.WORDPROC.REF.1.0 at all targets as follows:

```
Global name:      EURO.WORDPROC.REF.1.US

    Target:      FREDSWS
    Status:      Available

    Target:      JOHNSWS
    Status:      Installed, removable, active

Global name:      EURO.WORDPROC.REF.2.US

    Target:      FREDSWS
    Status:      Installed, removable, active
```

Related Information

The delcm command.

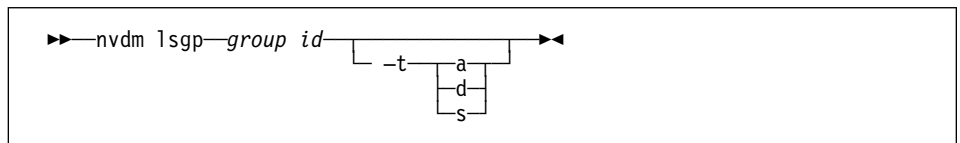
lsgrp – List Target Groups

Use the **lsgrp** command to report the configuration of a group or all groups. Output is listed for both static and dynamic groups. If a single group is specified, all of the member targets are listed. If all groups are selected, just the names and descriptions of all the groups are returned. You must have at least View or Modify Configuration authorization to use this command.

Because a single-node system has only one local target, the server itself, it has no groups, and this command is not available.

This command cannot be used locally by mobile clients.

Syntax



Parameters

group id

Group to be listed. You can use an asterisk (*) enclosed in single or double quotation marks to list all targets.

-t a|d|s

The type of group to be listed:

- a** All (both static and dynamic)
- d** Dynamic
- s** Static

Output

The output of the **lsgrp** command depends on whether a particular group is specified. If it is, output is as follows:

```

Group:      [group_id]
Mode:       [Push|Pull|Manager|Focal]
Type:       [Dynamic|Static]
Description: [group_description]

            [tgt_id_1] [description_1]
            [tgt_id_2] [description_2]
            .          .
            .          .
            [tgt_id_n] [description_n]
  
```

For the group ID, you can use an asterisk (*) enclosed in single or double quotation marks to list all groups. The output is as follows:

```
Group:      [group_id]
Mode:       [Push]
Type:       [Dynamic]
Description: [group_description]
```

```
Group:      [group_id]
Mode:       [Pull]
Type:       [Dynamic]
Description: [group_description]
```

```
Group:      [group_id]
Mode:       [Focal]
Type:       [Static]
Description: [group_description]
```

Examples

```
nvdm lsgp WORD
```

This command lists all targets in the group WORD. The output is as follows:

```
Group:      WORD
Mode:       Pull
Type:       Dynamic
Description: Word Processor Users

      FREDSWS      Fred's Workstation
      JOHNSWS      John's Workstation
      BACKUP        Backup Workstation
```

Related Information

The addgp, addpm, delgp, and delpm commands.

lspln

lspln – List a Plan

Use the **lspln** command to list the contents of a plan.

Syntax

►►—nvdm lspln—*global name*◄◄

Parameters

global name

The name of the plan. You can use wildcard characters, but a unique match must be found.

Examples

```
nvdm lspln SAMPLE.PLAN
```

This command produces the following output:

```
GLOBAL NAME:      SAMPLE.PLAN
DESCRIPTION:      Plan to learn about plans
DATA ACCESS KEY:  SAMPLE.PLAN
ERROR SEVERITY:   12
RECURSION TYPE:   NO_RECURSION
ENTRY:
  ID:             $00000
  FUNCTION:        SEND CHANGE.FILE.REF.1.US -Z 0 -S U -R U
  DESTINATION:     FREDSW
ENTRY:
  ID:             $00001
  FUNCTION:        SEND CHANGE.FILE.REF.1.US -Z 0 -V A
  DESTINATION:     FREDSW
  CONDITION:       EE($00000)<ERROR
ENTRY:
  ID:             $00002
  FUNCTION:        AUTH CHANGE.FILE.REF.2.US -Z 0
  DESTINATION:     FREDSW
```

Related Information

The **addpln**, **execpln**, and **updpln** commands.

Isprf – List Profiles

Use the **Isprf** command to list the contents of one or all authorization profiles. You must have View System Administration authorization to use this command, unless the profile is not associated with any user or the profile is your own.

Syntax

```
nvdm lsprf [profile name] [-l]
```

Parameters

profile name

The name of the authorization profile. If this parameter is specified, only that profile is displayed. If it is not specified, only the profile assigned to the user issuing the command is displayed. You can use an asterisk (*) enclosed in single or double quotation marks to list all authorization profiles.

- l The profile is to be listed in long format. The long format includes detailed information for each profile.

Output

The output of the **Isprf** command is as follows:

```
Profile Name:                <Profile name>
  Manage All Targets:        <Yes|No>
  Change Management Install: <Yes|No>
  Change Management Activate: <Yes|No>
  Change Management Execute: <Yes|No>
  Change Management Authorize: <Yes|No>
  Preparation:               <None|View|Modify>
  Send:                      <Yes|No>
  Retrieve, Delete and Replace: <Yes|No>
  Queues:                   <None|View|Manage>
  System Administration:    <None|View|Modify>
  Configuration:           <None|View|Modify>
  Purge Requests:          <Yes|No>
  Manage All Requests:      <Yes|No>
```

Examples

```
nvdm lsprf *
```

This command produces the following output:

```
Profile Name:                FNDADMN
Profile Name:                FNDBLD
Profile Name:                FNDUSER
```

lsprf

```
nvdn lsprf FNDADMN -l
```

This command produces the following output:

Profile Name:	FNDADMN
Manage All Targets:	Yes
Change Management Install:	Yes
Change Management Activate:	Yes
Change Management Execute:	Yes
Change Management Authorize:	Yes
Preparation:	Modify
Send:	Yes
Retrieve, Delete and Replace:	Yes
Queues:	Manage
System Administration:	Modify
Configuration:	Modify
Purge Requests:	Yes
Manage All Requests:	Yes

Related Information

The updprf command.

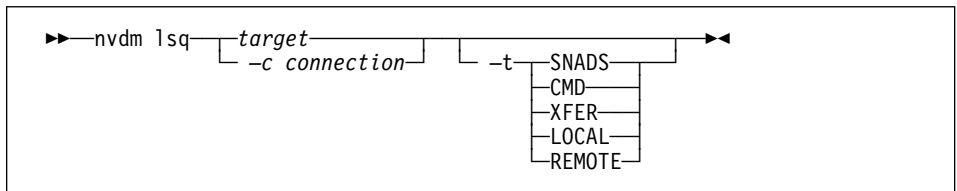
lsq – Display Queue Entry Details

Use the **lsq** command to display details about all entries on a queue. The details displayed are:

- The global name of the file being distributed
- The destination target and its network address
- The current status of the entry, which can be:
 - Not started
 - Scheduled
 - Paused
 - Pending acceptance
 - Ready
- The origin of the request
- The distribution ID

This command can be used locally by mobile clients. However, no parameters can be used when issuing it.

Syntax



Parameters

target

Name of the target the queue refers to. If the queue is on a local target, it is the queue name; if the queue is on a remote target, this parameter is the name of the next server connected.

If you specify a local target, you can omit the **-t** parameter. The system recognizes local targets and replaces the correct value for the type of queue. You can use wildcard characters, but a unique match must be found.

-c connection

The name of the connection the queue refers to. This is the name of the file where the connection is specified, in the `/usr/lpp/netviewdm/db/snadscon` directory.

This parameter applies only to queues on remote targets, because local queues are not associated with any connections.

You cannot use wildcard characters.

-t SNADS|CMD|XFER|LOCAL|REMOTE

The type of queue. It can be one of the following:

SNADS On SNA/DS connections

lsq

CMD On server-to-server (STS) connections to transfer commands
XFER On server-to-server (STS) connections to transfer files
LOCAL Local
REMOTE Remote

Output

The output of the **lsq** command is as follows:

```
Distribution ID:      [Dist ID]

Command:             [Command]
Sequence number:     [Seq number]
Originator:          [Originating target]
Submitted on:        [Submission date]
Global Name:         [Global name]
Status:              [Not started/Scheduled/Interrupted/
                     Waiting retry/Pending acceptance/Ready]
Request ID:          [Req ID]
```

Where:

- Distribution ID is a number identifying the entry
- Command is the command issued with the request
- Sequence number is a sequential number for each entry in a plan whose execution is in the queue
- Originator is the target that submitted the request
- Submitted on is the date when the request was submitted
- Global name is the global name of the file referred to by the command
- Status is the status of the request
- Request ID is the request identifier

Examples

```
nvdm lsq ROMA
```

This command displays the contents of the queue called ROMA. The output is as follows.

```
Distribution ID:      1

Command:             Install
Sequence number:     1
Originator:          FREDSW
Submitted on:        06/27/00
Global name:         CHANGE.FILE.REF.1
Status:              Ready
Request ID:          fredsws root 47 0
```

Distribution ID: 2

Command: Accept
Sequence number: 1
Originator: FREDSW
Submitted on: 06/27/00
Global name: CHANGE.FILE.REF.1
Status: Not started
Request ID: fredsws root 48 0

Distribution ID: 3

Command: Activate
Sequence number: 1
Originator: JOHNSWS
Submitted on: 06/27/00
Global name: ANOTHER.FILE.REF.1
Status: Not started
Request ID: johnsws root 55 0

Related Information

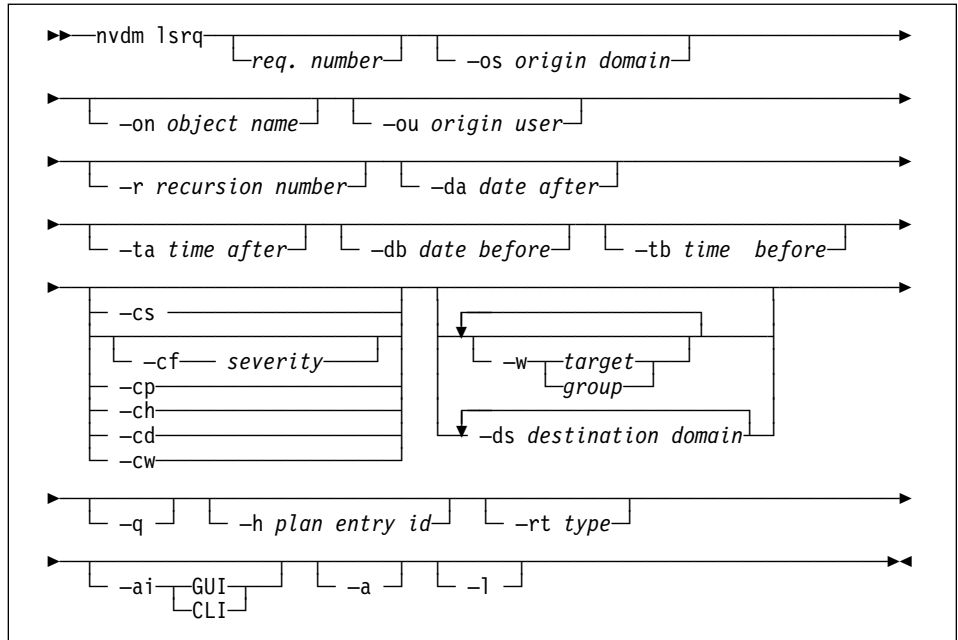
The stat, hldq, relq, and prgq commands.

lsrq – List Change Control and Distribution Requests

Use the **lsrq** command to list the status of requests that have been submitted. You can request information about a plan, plan entries or single requests that address a specific domain, or plan entries or single requests that address a specific target.

This command cannot be used at mobile clients when working with the local catalog.

Syntax



Parameters

request number

The sequential number of the request. You can use an asterisk (*) enclosed in single or double quotation marks to specify all requests.

-os *origin domains*

The domain where the requests to list originated. You can use the wildcard characters '*' and '?'.

-on *object name*

The global name of the object for which the users submitted the requests to list. You can use the wildcard characters '*' and '?'.

-ou *origin user*

The user who submitted the requests to list. Only the administrator can list requests submitted by other users. You can use the wildcard characters '*' and '?'. The default is the name of the user submitting this command.

-r *recursion number*

A specific instance of a recursive request. If you do not specify **-r**, the last instance only is displayed.

You can use an asterisk (*) enclosed in single or double quotation marks to display all the instances of a recursive request.

-da *date after*

List only those requests scheduled after this date.

-ta *time after*

List only those requests scheduled after this time.

-db *date before*

List only those requests scheduled before this date.

-tb *time before*

List only those requests scheduled before this time.

-cs

List only those requests whose execution completed successfully.

-cf

List only those requests whose execution failed. Requests can be filtered for a specific error severity. You can use an asterisk (*) enclosed in single or double quotation marks to list all failed requests.

-cp

List only those requests that are pending.

-ch

List only those requests that are held.

-cd

List only those requests that have been deleted.

-cw

List only those requests that are waiting.

-w *target*

The targets or groups that the requests listed address. You can use an asterisk (*) enclosed in single or double quotation marks to specify all targets. See "Entering Target Names" on page 12.

-ds *destination domain*

The domains the requests listed address. You can specify up to 16 domains. The wildcard characters '*' and '?' can be used.

-q

List the requests submitted to the product but not yet written to the product database. You can use this parameter if the request handler has not been started or is busy with other requests.

If **-q** is specified, the following parameters are ignored:

-h, -da, -ta, -db, -tb, -cs, -cf, -cp, -ch, -cd, -cw, -a

lsrq

-h *plan entry id*

The identification of the plan entries to be listed. If the request does not refer to a plan, an error message is displayed. You can use the wildcards '*' and '?'.

-rt *type*

The type of request to be listed (for example, inst, rem, or send). The default is all.

-ai **GUI|CLI**

List all the requests with the specified application identifier. You can use the wildcards '*' and '?'. Specify one of the following:

GUI

To list all the requests submitted using the Graphical User Interface (GUI)

CLI

To list all the requests submitted using the Command Line Interface (CLI)

- a** The request is listed in advanced format. Advanced format includes information about the servers and the targets involved in a request. If you do not specify this parameter, summary information is listed.

If the **-ds** parameter is specified, summary information on the targets is displayed.

If the **-w** parameter is specified, this parameter is ignored.

- l** The request is listed in long format. The long format includes detailed information for each request.

If the **-w** or **-ds** parameter is specified, this parameter is ignored.

Examples

The following command requests a list in the long format for the request target01.root.13.1:

```
nvdm lsrq 13 -l
```


It produces the following output:

```

Request ID:           target01 root 13 1
SNA correlator:       target01 07/05/00 2
Submission time:      07/05/00 12:03:14
Request type:         Install
Object:               EURO.WORDPROC.REF.1.US
Status:               Waiting
Error severity:       0
Schedule time:        07/29/00 12:03:14
Starting mode:        Released
Priority:              No
Monthly schedule:     10,20,30
Max severity:         0
Stop on instance overlap: No
Recursion expiration: Never
Application ID:        GUI
Execution window:
  Execution time :     When received by target
  Expiration time:     N/A - 23:59:59
Time Format :          Local time at origin
Termination target exit:
Termination exit:

```

The following command requests a list of all plan entries for the plan request target03.root.11.0:

```
nvdms lsrq 11 -h '*' -l
```

It produces the following output:

```

Request ID:           target03 root 11 0 $00000
SNA correlator:       target03 07/13/00 53
Submission time:      07/13/00 17:51:50
Request type:         send
Object:               EURO.WORDPROC.REF.1.US
Status:               Successful
Error severity:       0

Request ID:           target03 root 11 0 $00001
SNA correlator:       target03 07/13/00 54
Submission time:      07/13/00 17:51:50
Request type:         Install
Object:               EURO.WORDPROC.REF.1.US
Status:               Scheduled
Error severity:       0
Execution mode:        Normal

```

The following command requests the long format for the plan entry \$00001 of the plan request target03.root.11.0.

```
nvdms lsrq 11 -h '$00001' -l
```

It provides the following output:

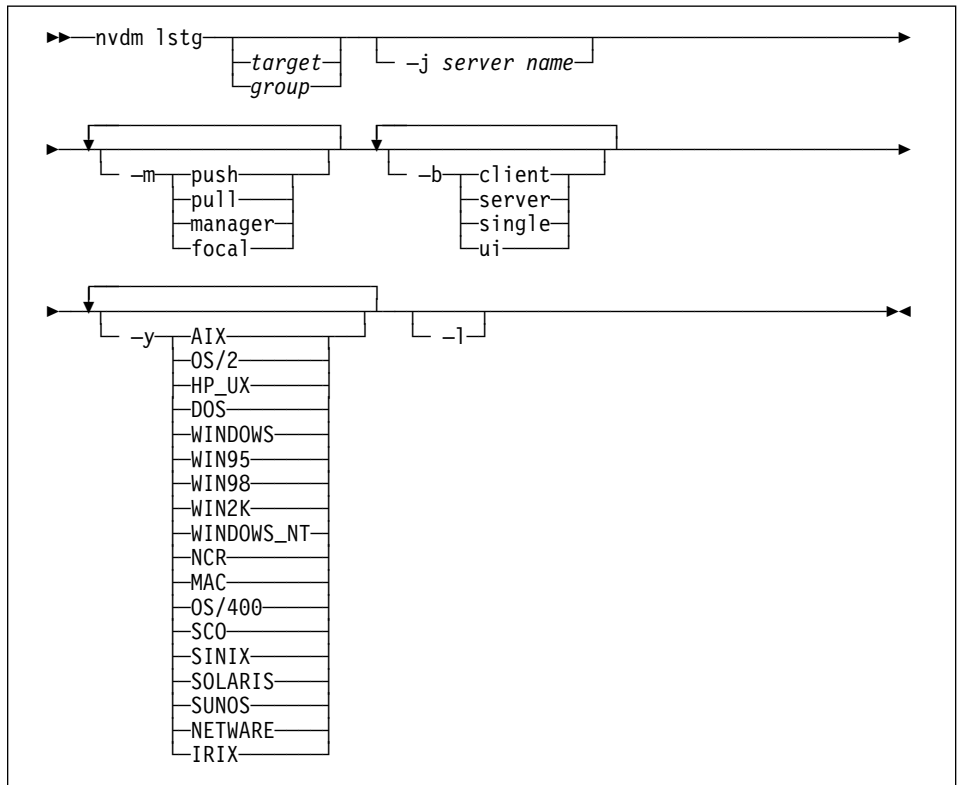
```
Request ID:          target03 root 11 0 $00001
SNA correlator:      target03 07/04/00 53
Submission time:     07/04/00 17:51:50
Request type:        Install
Object:              EURO.WORDPROC.REF.1.US
Status:              Scheduled
Error severity:      0
Schedule time:       07/04/00 17:51:50
Starting mode:       Released
Priority:             No
Application ID:       GUI
Execution window:
  Execution time :    When received by target
  Expiration time:    Undefined
Time Format:          Local time at origin
Condition:
Execution mode:       Normal
```

lstg – List Targets

Use the **lstg** command to report the configuration of one or more targets. You must have at least View or Modify Configuration authorization to use this command.

This command cannot be used at mobile clients when working with the local catalog.

Syntax



Parameters

target | group

The targets or groups to be listed. You can enter up to 16 names of targets or groups. This parameter is optional. You can use an asterisk (*) enclosed in single or double quotation marks to list all targets and groups, but you can restrict the listing using the **-m**, **-b**, and **-y** parameters.

-j server name

Specify this parameter only if you specify *client* for the **-b** parameter or if the **-b** parameter was not specified. All clients belonging to the server specified are listed.

-m push|pull|manager|focal

The target mode.

-b client|server|single|ui|mobile

The target type.

-y AIX|OS/2|HP_UX|DOS|WINDOWS|WIN95|WIN98|WIN2K|WINDOWS_NT|SCO|NCR|MAC|SINIX|SOLARIS|SUNOS|NETWARE|IRIX|OS/400

The target operating system. Note that WINDOWS indicates Windows 3.11, while WIN95, WIN98 and WIN2K are Windows 95, Windows 98 and Windows 2000, respectively.

-l List the information in long mode, which includes:

- Target
- Description
- Customer name
- Contact name
- Telephone number
- Manager
- Mailing address
- Target access key
- Mode
- Server name
- Type
- Operating system
- Target address
- Domain address
- LAN address
- CM window
- Distribution window
- Network
- Logging level
- Tracing state
- Installation parameters
- Hardware parameters
- Discovered inventory

If the parameter is omitted, only the target, mode, description, and type are listed for each target.

Output

The output of the **lstg** command can be in one of two formats, depending on the setting of the **-l** parameter. The short (**-l** parameter omitted) format is as follows:

```
Target:      [ws_id]
Mode:        [Mode]
Description: [Description]
Type:        [Type]
```

The long format is as follows.

```

Target: [ws_id]
Description: [description]
Customer name: [customer name]
Contact name: [contact name]
Telephone number: [telephone number]
Manager: [owning manager]
Mailing address: [address]
               [address]
               [address]
Target Access Key: [TAK]
Mode: [Push|Pull|Manager|Focal]
Server Name: [server name]
Type: [Client|Server|Single|UI]
Operating system: [AIX|OS/2|HP_UX|WINDOWS|WINDOWS_
                  NT|DOS|SOLARIS|SUNOS|SCO|NCR|MAC|
                  SINIX|NETWARE|IRIX|OS/400]

Target address: [tg_address]
Domain address: [dm_address]
LAN address: [LAN address]
CM window: [start_time] - [stop_time]
Distribution window: [start_time] - [stop_time]
Network: [TCP hostname|APPC netid.lname.lumode or profile|
          UNKNOWN]

Logging level: [Minimal|Normal|Diagnostic]
Tracing state: [On|Off]
Installation parms: [parm=value]
                  [parm=value]
Shared tokens: [parm=value]
               [parm=value]
Hardware parms: [parm=value]
               [parm=value]
Discovered inventory: [parm=value]
                    [pram=value]

```

The last four items (installation parameters, hardware parameters, discovered inventory, and users) are each repeated as many times as necessary. If none is defined, these lines are reported as follows.

```

Installation parms: (none)
Hardware parms: (none)
Discovered inventory: (none)
Users: (none)

```

Examples

```
nvdms 1stg FREDSW
```

This command lists the target FREDSW in short mode. The output looks like the following:

Istg

```
Target:          FREDSW
Mode:            Pull
Description:     Fred's Computer
Type:           CLIENT
```

```
nvdm lstg '*' -l
```

This command lists all targets in long mode. The output looks like the following.

```
Target:          FREDSW
Description:     Fred's Computer
Customer name:   AmCar
Contact name:    Fred Smith
Telephone number: 011-222-3333
Manager:         Martyn Jones
Mailing address: Company Offices
                 The AmCar Company
                 Baltimore

Target access key: TAK3
Mode:            Push
Server name:     BARRY
Type:            Client
Operating system: Platform operating system
Target address:  AmCar target
Domain address:  NTWK1.SALES
LAN address:     10:0:5a:3a:2a:8b.
CM window:       00:00 - 23:59
Distribution window: 00:00 - 23:59
Network:         TCP p57ri0d
Logging level:   Diagnostic
Tracing state:   Off
Installation parms: base=/usr/lpp/apps/nvdm
                  word=/usr/lpp/apps/word
Hardware parms:  (none)
Discovered Inventory: (none)
```

Related Information

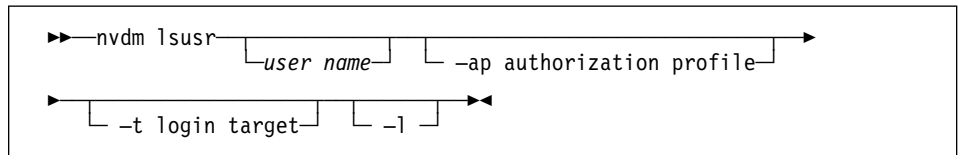
The addtg, deltg, updtg, stattg, and lsgp commands.

lsusr – List User Authorizations

Use the **lsusr** command to list the authorization defined for each user.

You must have View or Modify System Administration authorization to use this command.

Syntax



Parameters

user name

The name of the user. If user name is not specified, the command applies to the user issuing the command. You can use an asterisk (*) enclosed in single or double quotation marks to indicate all users.

-ap *authorization profile*

Display all users that have an authorization profile specified.

-t *login target*

Display all users with login authorization for the target specified. The target must be a local target.

-l Display user information in the long form.

Examples

```
nvdm lsusr james -ap builder -l
```

This command lists the authorization for user james who has the builder authorization profile. The information is listed in the long format.

```

User:                james
Authorization Profile:  builder
Description:
Targets allowed to login at:  Yes
Target excluded to login to:  nvdm14
Target excluded to login to:  nvdm16

```

```

User:                marcus
Authorization Profile:  user
Description:
Targets allowed to login at:  Yes
Target allowed to login to:  nvdm14
Target allowed to login to:  nvdm16

```

lsusr

Related Information

The `delusr` and the `updusr` commands.

nvdm – Start a Command Line Session

Use this command to start an interactive command line session. Enter:

```
nvdm
```

You are prompted to enter your user ID and password, if any.

After the command is successfully executed, a new prompt is displayed to notify you that the interactive mode is active:

```
nvdm>
```

After starting a command line session, you can enter commands without specifying the prefix *nvdm*. The following commands are also available:

- open** Use this command to open a connection to a TME 10 Software Distribution server while you are in an interactive command line session. For more details, refer to the open command.
- quit** Use this command to close an interactive command line session. For more details, refer to the quit command.
- !** Use this command to move to the command line of your operating system, keeping the interactive command line session still open. Enter *exit* at the operating system command line to return to the interactive command line session that is kept open.
- ?** Use this command to display help information relating to use of the command-line interface. Enter **?** alone to display a list of all the commands. Enter **?** followed by a specific command name to display help information about that command.

open

open – Open a Connection to a TME 10 Software Distribution Server

Use the **open** command to open a connection to a TME 10 Software Distribution server while you are in an interactive command line session. When you issue this command, you are prompted to specify your user ID and password. You then no longer need to supply this information for each command you enter.

Syntax



Parameters

server id

The name of the server to connect to.

Examples

```
open alpha
```

This command opens a session with server alpha.

Related Information

The **nvd**, **svr**, and **quit** commands.

preqpln

If you specified `inst` as the request, the option `-i` (to force the installation to be scheduled even if the change file history on the target does not allow it) is to be included in this list. If you do not, this option is used anyway.

- s** Specify this parameter to have the change file and its prerequisites sent to the servers of the targets. This is in addition to the type of request specified with the **-rt** parameter. This parameter makes sense if your request addresses to targets in remote domains or if you specified `inst` for the **-rt** parameter.

Specify **-so** to define a list of options for the **-s** parameter. If you specified `inst` or `auth` as the request, the option `'-o'` (to overwrite the existing file) is to be included in this list. If you do not, this option will be used anyway.

Examples

```
nvdn preqpln file1.ref.1 plan.pro -rt inst -s
```

This command generates a plan profile (saved as `plan1.pro`) that has the following entries to:

- Send each prerequisite file to the servers of targets
- Send the change file `cf1.ref.1` to the servers of targets
- Install each prerequisite file to the targets
- Install the change file `cf1.ref.1` to the targets

Prerequisite files are installed in ascending order; that is, the last required is installed first.

Output

The output of the **preqpln** command is a plan profile that you can save in a text file. For example, the command:

```
nvdn preqpln file1.ref.1 plan.pro -rt inst -s
```

creates a plan profile named `plan.pro` that looks like the following:

GLOBAL NAME: GENERATED.PLAN.SERVER.ROOT.2000M02D27.H12M31S15.3
 DESCRIPTION: Generated Plan to INST prerequisites of CF file1.ref.1

ENTRY:
 ID: \$00001
 FUNCTION: SEND PREREQ2.REF.1 -0
 DESTINATION: \$(SERVERLIST)

ENTRY:
 ID: \$00002
 FUNCTION: SEND PREREQ1.REF.1 -0
 DESTINATION: \$(SERVERLIST)
 CONDITION: ET(\$00001) <= WARNING

ENTRY:
 ID: \$00003
 FUNCTION: SEND FILE1.REF.1 -0
 DESTINATION: \$(SERVERLIST)
 CONDITION: ET(\$00002) <= WARNING

ENTRY:
 ID: \$00004
 FUNCTION: INST PREREQ2.REF.1 -I
 DESTINATION: \$(TARGETLIST)
 CONDITION: ED(\$00003) <= WARNING

ENTRY:
 ID: \$00005
 FUNCTION: INST PREREQ1.REF.1 -I
 DESTINATION: \$(TARGETLIST)
 CONDITION: ET(\$00004) <= WARNING

ENTRY:
 ID: \$00006
 FUNCTION: INST FILE1.REF.1 -I
 DESTINATION: \$(TARGETLIST)
 CONDITION: ET(\$00005) <= WARNING

Related Information

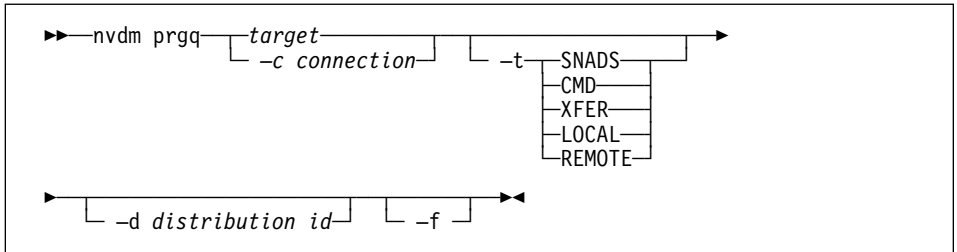
The addpln and execpln commands.

prgq – Purge One or All Entries from a Queue

Use the **prgq** command to purge one or all entries from a queue. By default, confirmation is requested before the command is performed.

This command can be used locally by mobile clients. Only the **-d** and **-f** parameters can be used when issuing it.

Syntax



Parameters

target

The name of the target that the queue refers to. If the queue is on a local target, this parameter is the queue name; if the queue is on a remote target, this parameter is the name of the next server connected.

You cannot use wildcard characters.

This parameter cannot be used locally by mobile clients.

-c connection

The name of the connection the queue refers to. This is the name of the file where the connection is specified, under the `/usr/lpp/netviewdm/db/snadscon` directory.

This parameter applies only to queues on remote targets, as local queues are not associated to any connections.

You cannot use wildcard characters.

-t SNADS|CMD|XFER|LOCAL|REMOTE

The type of queue. It can be:

SNADS On SNA/DS connections

CMD On server-to-server (STS) connections to transfer commands

XFER On server-to-server (STS) connections to transfer files

LOCAL Local

REMOTE Remote

-d distribution id

Identifier of the entry to purge, obtained from the **lsq** command. If this is omitted, all entries on the queue are purged. You can also use this parameter to purge a

request that is currently executing on a local target. Confirmation is requested twice.

- f** Confirmation is not requested before purging the queue. If **-f** is omitted, you are prompted to type *y* to confirm that the queue should be purged.

Examples

```
nvdn prgq DETROIT
```

This command purges all distributions on the queue named DETROIT. Before the command is carried out, confirmation is requested.

Related Information

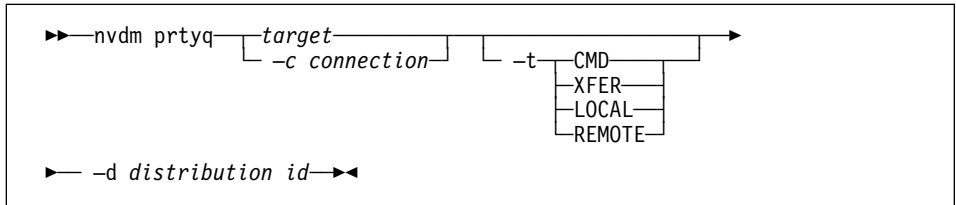
The `lsq`, `hldq`, and `relq` commands.

pptyq – Give Priority to a Queue

Use the **pptyq** command to give priority to an entry in a queue. The entry you specify is moved to the top of queue and processed immediately after the current request is processed. If a second entry is subsequently given priority, it is processed first.

If the system is restarted, the new priority is ignored. The queue is generated in its original order.

Syntax



Parameters

target

The name of the target that the queue refers to. If the queue is on a local target, it is the name of queue; if the queue is on a remote target, it is the name of the next server connected.

You cannot use wildcard characters.

-c connection

The name of the connection the queue refers to. This is the name of the file where the connection is specified, under the /usr/lpp/netviewdm/db/snadscon directory.

This parameter applies only to queues on remote targets, as local queues are not associated to any connections.

You cannot use wildcard characters.

-t CMD|XFER|LOCAL|REMOTE

The type of queue. It can be respectively the queue:

- On server-to-server (STS) connections to transfer commands
- On server-to-server (STS) connections to transfer files
- Local
- Remote

-d dist. id

Identifier of the distribution to move up one position in the queue, obtained from the **lsq** command. This parameter is required.

Examples

```
prtyq FREDSW -d 3
```

This command gives priority to distribution 3 for target FREDSW.

Related Information

The lsq command.

quit

quit – Close an Interactive Command Line Session

Use the **quit** command to close an interactive command line session. The user ID and password entered to access the interactive session are no longer active.

Syntax

▶▶—quit—▶▶

Examples

quit alpha

This command closes the session with server alpha.

Related Information

The nvdm and the open command.

relc – Release Remote Communication

Use the **relc** command to release all remote communication, undoing the effect of the **hldc** command.

The command can be used to release communications:

- On SNA/DS connections
- On server-to-server (STS) connections to transfer commands
- On server-to-server (STS) connections to transfer files

Syntax



Parameters

SNADS|CMD|XFER

The communication type to be released. It can be one of the following:

SNADS On SNA/DS connections

CMD On server-to-server (STS) connections to transfer commands

XFER On server-to-server (STS) connections to transfer files

If none of these values is specified, all the remote communications are released.

Examples

```
nvdm relc
```

This command releases the SNA network communication to and from the server.

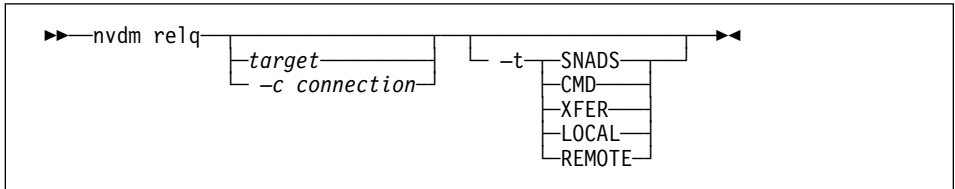
Related Information

The **hldc** command.

relq – Release One or All Queues

Use the **relq** command to release one or all communication queues.

Syntax



Parameters

target

The name of the target that the queue refers to. If the queue is on a local target, this parameter coincides with the queue name itself; if the queue is on a remote target, this parameter is the name of the next server connected.

You can use an asterisk (*) enclosed in single or double quotation marks to indicate all targets.

-c connection

The name of the connection the queue refers to. This is the name of the file where the connection is specified, in the /usr/lpp/netviewdm/db/snadscon directory.

This parameter applies only to queues on remote targets, because local queues are not associated with any connections.

You can use wildcard characters.

-t SNADS|CMD|XFER|LOCAL|REMOTE

The type of queue. It can be one the following:

SNADS On SNA/DS connections

CMD On server-to-server (STS) connections to transfer commands

XFER On server-to-server (STS) connections to transfer files

LOCAL Local

REMOTE Remote

This parameter is optional. If you do not specify it, all the remote queues are released.

Examples

```
nvdm relq DETROIT
```

This command releases the queue called DETROIT.

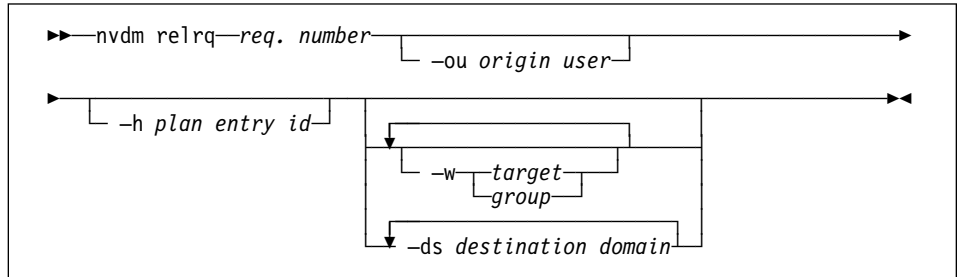
Related Information

The lsq, hldq, and prgq commands.

relrq – Release a Request

Use the **relrq** command to release transmission requests for targets addressed by a request, all the targets in a domain, a single target, or all the requests in a plan.

Syntax



Parameters

request number

The sequential number of the request. This parameter is required. You cannot use wildcard characters.

-ou *origin user*

The user who submitted the request. Only the administrator can release requests submitted by other users. This parameter is optional. The default is the name of the user submitting this command.

-h *plan entry id*

The identification of the plan entry to be released. If the request does not refer to a plan, an error message is displayed. If you do not specify this parameter and the request refers to a plan, the entire plan is released. If you do not specify either **-w** or **-ds**, all requests for this entry are held.

-w *target | group*

The targets or groups addressed by the request to be released. See “Entering Target Names” on page 12.

You cannot specify this parameter if you specify the **-ds** parameter.

-ds *destination domain*

The domain addressed by the request to be released. You cannot specify this parameter if you specify the **-w** parameter.

Examples

```
nvdm relrq 13 -w FREDSSWS
```

This command releases the request whose number is 13 for target FREDSSWS.

Related Information

The delrq, eraserq, and hldrq commands.

reltg – Release a Target

Use the **reltg** command to release a held local target or group.

Syntax



Parameters

target | group

Target or group to be released. This parameter is optional and follows the same rules for the workstation parameter. You can use an asterisk (*) enclosed in single or double quotation marks to release all targets and groups.

Examples

```
nrdm reltg FREDWS
```

This command releases the target named FREDWS.

Related Information

The `hldtg` and `stattg` commands.

Parameters

global name

Global name of the change file to remove. See “Entering Global Names” on page 13.

You can use wildcard characters, but the removal operation is scheduled only if a unique match is found.

-w *target | group*

The targets or groups to which this command applies. See “Entering Target Names” on page 12.

This parameter cannot be used locally by mobile clients.

-v *a|s*

Disk area in which to perform the Remove operation. If **-vs** is specified, the backup copies are copied to the service area and then copied to the active area on receipt of a subsequent activate request.

If **-va** is specified, the Remove operation is performed directly in the active area. If **-v** is not specified, **-va** is used as the default.

- i** Force the Remove command to be scheduled even if the change file history on the target does not allow it. However, the removal will not succeed if the change file history forbids it at the time when the removal is to be executed.

-d *schedule date*

The date the request is to take place at the origin target.

Date and time parameters are described in detail in “Entering Dates and Times” on page 13.

-t *schedule time*

The time the request is to take place at the origin target.

-db *request execution date*

The date the request is to take place at the destination target.

-tb *request execution time*

The time the request is to take place at the destination target.

-da *request expiration date*

The date after which the request cannot take place at the destination target.

-ta *request expiration time*

The time after which the request cannot take place at the destination target.

-dc

This parameter refers to operations performed at mobile clients. When it is specified, the operation is to be performed when the mobile client is disconnected from the server.

-hd

This parameter pertains to operations performed at mobile clients, and can only be specified together with the **-dc** parameter. It holds the disconnected request at the

server until its execution time is reached. The request is forwarded to the client during the next connection window opened after the execution time specified, and executed there when the client is disconnected.

-z old

The format of the execution time. Enter one of the following values:

- o The time is expressed as the local time at the origin target. This is the default.
- d The time is expressed as the local time at the destination target.

This parameter cannot be used locally by mobile clients.

-h The request is to be submitted with high priority. You cannot specify this parameter if schedule times and execution times are specified.

This parameter cannot be used locally by mobile clients.

-q The request is to be submitted in held state. The request is scheduled only when released by the operator.

This parameter cannot be used locally by mobile clients.

-l application id

The ID of the application used to submit the request. This ID can be used later to filter the requests submitted by a specific application.

It does not make sense to set an application ID for a request issued from the graphical interface, as this is an interactive process. Requests submitted from the graphical interface are automatically given the value of GUI as the application ID. You cannot change this value.

If you do not specify an application ID for requests issued from the command-line interface, the default value is CLI.

-rd rec. date interval

A recursion date interval expressed in years, months, days. The request is executed upon every lapse of this interval. To specify an interval in terms of days, specify also the months and the years. To specify an interval in terms of months, specify also the years.

For example, 1 means the request is executed every year; 0,1 means the request is executed every month; 0,2,1 means the request is executed every day of every other month.

This parameter cannot be used locally by mobile clients.

-rt rec. time interval

A recursion time interval expressed in hours and minutes. The request is executed upon every lapse of this interval. The format of a time interval is the same as for a time. For example, 3:45 means a time interval of 3 hours and 45 minutes.

This parameter cannot be used locally by mobile clients.

-rw *rec. week schedule*

The days of the week when the recursive request is to take place. The days of the week are as follows:

- 0 = Sunday
- 1 = Monday
- 2 = Tuesday
- 3 = Wednesday
- 4 = Thursday
- 5 = Friday
- 6 = Saturday

For example, to set a recursion for each Monday and Friday of the week, enter:

-rw 1,5

This parameter cannot be used locally by mobile clients.

-rm *rec. month schedule*

The days of the month when the recursive request is to take place. Each day of the month is represented by a number from 1 to 31.

For example, to select a recursion for the 15th and 27th of the month, enter:

-rm 15,27

The number 31 specifies the last day of any month. The number 29 specifies the last day of February even in a non-leap year.

This parameter cannot be used locally by mobile clients.

-rs *rec. max severity* **0|4|8|12|16**

A new occurrence of the request is generated only if the severity level returned from the previous occurrence is less than or equal to the max-severity specified.

Specify one of the following:

- 0** Successful
- 4** Warning
- 8** Error
- 12** Failed
- 16** Unrecoverable

This parameter cannot be used locally by mobile clients.

-ro Specify this parameter to stop the recursion mechanism if the current occurrence of the request is in progress when the new occurrence should be generated. If you do not specify this parameter, one or more occurrences may be skipped until the current one is completed, but the recursion mechanism is not stopped. This parameter is not set by default.

This parameter cannot be used locally by mobile clients.

-ed *rec. expiration date*

The date after which the system stops the recursion mechanism.

This parameter cannot be used locally by mobile clients.

-et *rec. expiration time*

The time after which the system stops the recursion mechanism.

This parameter cannot be used locally by mobile clients.

-m *target:procedure*

A cataloged procedure to execute on a given target when the request is completed. See "Running a Procedure after a Request Has Completed" on page 16.

This parameter cannot be used locally by mobile clients.

Examples

```
nvdms rem EURO.WORDPROC.REF.2.US -w FREDWS
```

This command removes the previously installed change file EURO.WORDPROC.REF.2.US from target FREDWS, with immediate effect but requiring activation.

Related Information

The acc, act, lrsq, inst, and uninst commands.

rentg – Rename a Target

Use the **rentg** command to rename a target in the product database. You can change the target name only if there are not pending requests for it. Local and remote targets can be renamed. The target is also renamed in any groups to which it belongs.

You can rename the local server, too. If you do so, then be sure to stop the server and all the clients and perform the following steps:

- Rename the physical node and add the new TCP/IP name to the TCP/IP configuration on the server.
- Edit the WORKSTATION NAME and SERVER keywords in the `/usr/lpp/netviewdm/nvdm.cfg` file to reflect the server's new name
- Erase all of the configuration files located in the `/usr/lpp/netviewdm/uicfg` directory
- Edit the name of the server in the `nvdm.cfg` on each client to reflect the new name of the server.
- Start the server and all the clients.

You must have full configuration authorization to use this command.

Syntax

```
nvdm rentg—old target—new target—[ -f ]
```

Parameters

old target

Existing target to be renamed. Must be specified precisely.

new target

New name for target. No existing target can have this name.

-f Confirmation is not requested before the rename is performed. If **-f** is omitted, you must confirm the rename.

Examples

```
nvdm rentg FREDWS JOESWS
```

This command renames the target FREDWS to JOESWS. Note that none of the parameters to the configuration of the target are changed. The description will remain the same and may be confusing. To change the configuration of a target, use the **updtg** command.

Related Information

The addtg, deltg, lstg, and updtg commands.

reset

reset – Reset TME 10 Software Distribution

Use the **reset** command to reset TME 10 Software Distribution, deleting all distributions and pending requests. Use this command only if TME 10 Software Distribution is not operating properly, and if the message log indicates that requests are not being read correctly from the queues. It can be issued only:

- By a user logged on to the workstation as root
- By a user that is defined as root to TME 10 Software Distribution
- From a server

Syntax

▶▶—nvdm reset—▶▶

Related Information

The start and stop commands.

rld – Reload Configuration Information

Use the **rld** command to reload configuration data in SNA/DS connection configuration files, SNA/DS connection files, and the routing table. You do this, for example, when new routes have been added to the routing table using an editor. You must have Modify Configuration authorization to use this command.

The base configuration file, `nvdn.cfg`, is not affected by the **rld** command. To make changes to this file operative, you must stop and then restart the product. If, however, the graphical interface or any other command is active when the stop command is issued, the reload operation is not successful. All tasks must be stopped before issuing the command.

Syntax

`►►—nvdn rld—◄◄`

Examples

```
nvdn rld
```

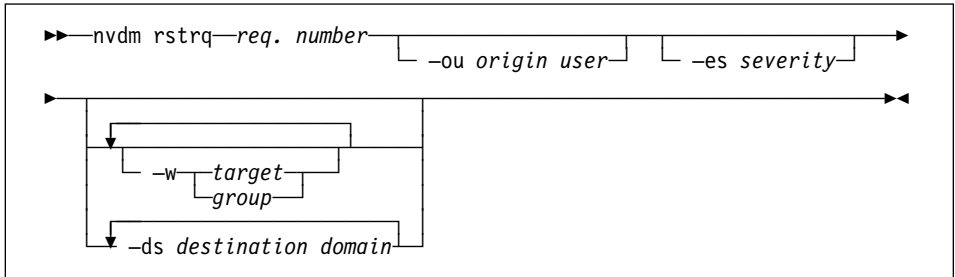
Causes remote connection configuration files to reload their configuration data.

rstrq – Restart a Request

Use the **rstrq** command to restart a request that failed for specific domains or targets.

You can not restart a failed instance of a recursive request.

Syntax



Parameters

request number

The sequential number of the request to be restarted.

-ou *origin user*

The user who submitted the request. Only the administrator can restart requests submitted by other users. This parameter is optional. The default is the name of the user submitting this command.

-es *severity*

The error severity for which the request has to be restarted. Specify a value equal to or higher than 8. The default is 8.

-w *target | group*

The targets or groups where the request is to be executed. See “Entering Target Names” on page 12. This parameter cannot be specified if **-ds** is specified.

-ds *destination domain*

Specifies the domain for which the request is to be restarted. You cannot specify this parameter if **-w** is specified.

Examples

```
nvdms rstrq 13 -w FREDWS
```

This command restarts the request whose number is 13 for target FREDWS.

Related Information

The **delrq**, **eraserq**, **hldrq**, and **relrq** commands.

rtrv – Retrieve a File

Use the **rtrv** command to retrieve a file, identified by its global file, from another target. The file can be retrieved to your workstation or to the server. If the global name that you specify does not yet exist on your server, then the file is cataloged when it arrives; otherwise, it uses the existing catalog entry (which must not refer to an existing local file).

You can transmit and store files in compressed or decompressed format across SNA or TCP/IP network links. You cannot change the compression format when transmitting within your domain, because the source and destination workstations share the same catalog and the compression format of the file must remain the same on each.

You can choose whether the transmitted file is to be compressed or decompressed at local server or destination target. Set the ACT_ON_TARGET environment variable by entering the following command at a command line:

```
set ACT_ON_TARGET=0 | 1
```

where 0 and 1 have the following meaning:

- 0** When a remote server sends a file in compress format and wants to store it in decompress format at destination target, the file is decompressed at local server and sent in decompress format to destination target.

When a remote server retrieves a file in compress format the file is compressed at local server and sent in compress format to the destination server.

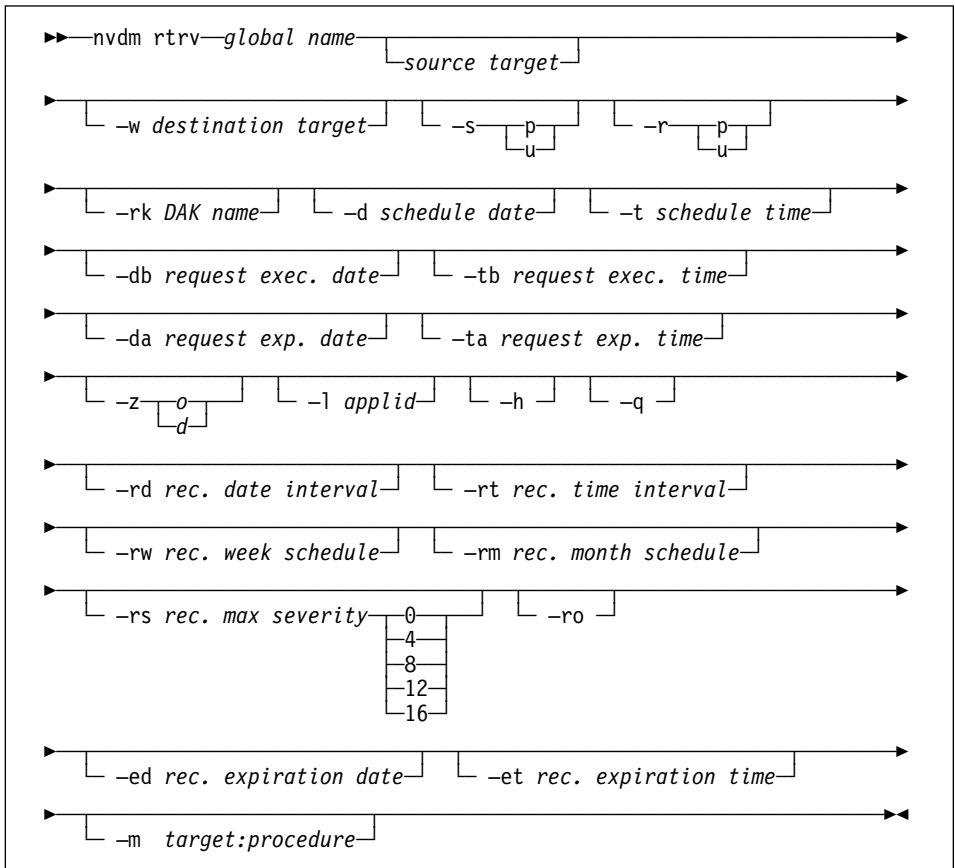
- 1** When a remote server sends a file in compress format and wants to store it in decompress format at destination target, the file is sent in compress format and it is decompressed at destination target.

When a remote server retrieves a file in compress format, the file is compressed at source target and transmitted as compressed.

For targets that have installed TME 10 Software Distribution versions prior to 3.1, this variable can be set only to 0. The default is 0.

This command cannot be used at mobile clients when working with the local catalog.

Syntax



Parameters

global name

The global name of the file. For files that are being retrieved, you cannot use the wildcard characters H and L in the global name. See “Entering Global Names” on page 13.

You can use wildcard characters, but a unique match must be found on the remote target.

source target

The target from which the file is to be retrieved. If this is not specified, the file is retrieved from the focal point.

If no focal point is configured, you must enter a source.

-w *destination target*

The target on which the retrieved file is to be placed (must be the local target or the server). If no **-w** parameter is specified, the file is saved on your workstation.

-s p|u (*sender's action*)

Transmit the file in packed (compressed) or unpacked (decompressed) format.

-sp File is to be packed

-su File is to be unpacked

If **-s** is omitted, no packing or unpacking is done before transmitting the file.

You cannot specify this parameter for retrieval operations from focal points running NetView DM Release 1.5.

-r p|u (*receiver's action*)

Store the file in packed (compressed) or unpacked (decompressed) format after it is retrieved.

-rp File is to be packed

-ru File is to be unpacked

If **-r** is omitted, no packing or unpacking is done after receiving the file.

-rk *DAK name*

The file is retrieved with the data access key specified. If **-rk** is not specified, the file is retrieved with the data access key specified in the catalog.

-d *schedule date*

The date the request is to take place at the origin target.

Date and time parameters are described in detail in "Entering Dates and Times" on page 13.

-t *schedule time*

The time the request is to take place at the origin target.

-db *request execution date*

The date the request is to take place at the destination target.

-tb *request execution time*

The time the request is to take place at the destination target.

-da *request expiration date*

The date after which the request cannot take place at the destination target.

-ta *request expiration time*

The time after which the request cannot take place at the destination target.

-z o|d

The format of the execution time. Enter one of the following values:

o The time is expressed as the local time at the origin target. This is the default.

d The time is expressed as the local time at the destination target.

-l *application id*

The ID of the application used to submit the request. This ID can be used later to filter the requests submitted by a specific application.

It does not make sense to set an application ID for a request issued from the graphical interface, because this is an interactive process. Requests submitted from the graphical interface are automatically given the value of GUI as the application ID. You cannot change this value.

If you do not specify an application ID for requests issued from the command-line interface, the default value is CLI.

- h** The request is to be submitted with high priority. You cannot specify this parameter if schedule times and execution times are specified.
- q** The request is to be submitted in held state. The request is scheduled only when released by the operator.

This parameter cannot be used locally by mobile clients.

-rd *rec. date interval*

A recursion date interval expressed in years, months, days. The request is executed upon every lapse of this interval. To specify an interval in terms of days, specify also the months and the years. To specify an interval in terms of months, specify also the years.

For example, 1 means the request is executed every year; 0,1 means the request is executed every month; 0,2,1 means the request is executed every day of every other month.

-rt *rec. time interval*

A recursion time interval expressed in hours and minutes. The request is executed upon every lapse of this interval. The format of a time interval is the same as for a time. For example, 3:45 means a time interval of 3 hours and 45 minutes.

-rw *rec. week schedule*

The days of the week when the recursive request is to take place. The days of the week are as follows:

- 0 = Sunday
- 1 = Monday
- 2 = Tuesday
- 3 = Wednesday
- 4 = Thursday
- 5 = Friday
- 6 = Saturday

For example, to set a recursion for each Monday and Friday of the week, enter:

```
-rw 1,5
```

-rm *rec. month schedule*

The days of the month when the recursive request is to take place. Each day of the month is represented by a number from 1 to 31.

For example, to select a recursion for the 15th and 27th of the month, enter:

```
-rm 15,27
```

The number 31 specifies the last day of any month. The number 29 specifies the last day of February even in a non-leap year.

-rs *rec. max severity* **0|4|8|12|16**

A new occurrence of the request is generated only if the severity level returned from the previous occurrence is less than or equal to the max-severity specified.

Specify one of the following:

- 0** Successful
- 4** Warning
- 8** Error
- 12** Failed
- 16** Unrecoverable

-ro Specify this parameter to stop the recursion mechanism if the current occurrence of the request is in progress when the new occurrence should be generated. If you do not specify this parameter, one or more occurrences may be skipped until the current one is completed, but the recursion mechanism is not stopped. This parameter is not set by default.

-ed *rec. expiration date*

The date after which the system stops the recursion mechanism.

-et *rec. expiration time*

The time after which the system stops the recursion mechanism.

-m *target:procedure*

A cataloged procedure to execute on a given target when the request is completed. See "Running a Procedure after a Request Has Completed" on page 16.

Examples

```
nvdms rtrv EURO.WORDPROC.UPD.2.3.US JOHNSWS -sp -ru
```

This command causes the file EURO.WORDPROC.UPD.2.3.US to be retrieved from JOHNSWS. The file is to be sent in a compressed format and decompressed on receipt. The algorithm specified in the catalog entry is used for the compression and decompression. The file is retrieved to your computer.

Related Information

The send, cat, and del commands.

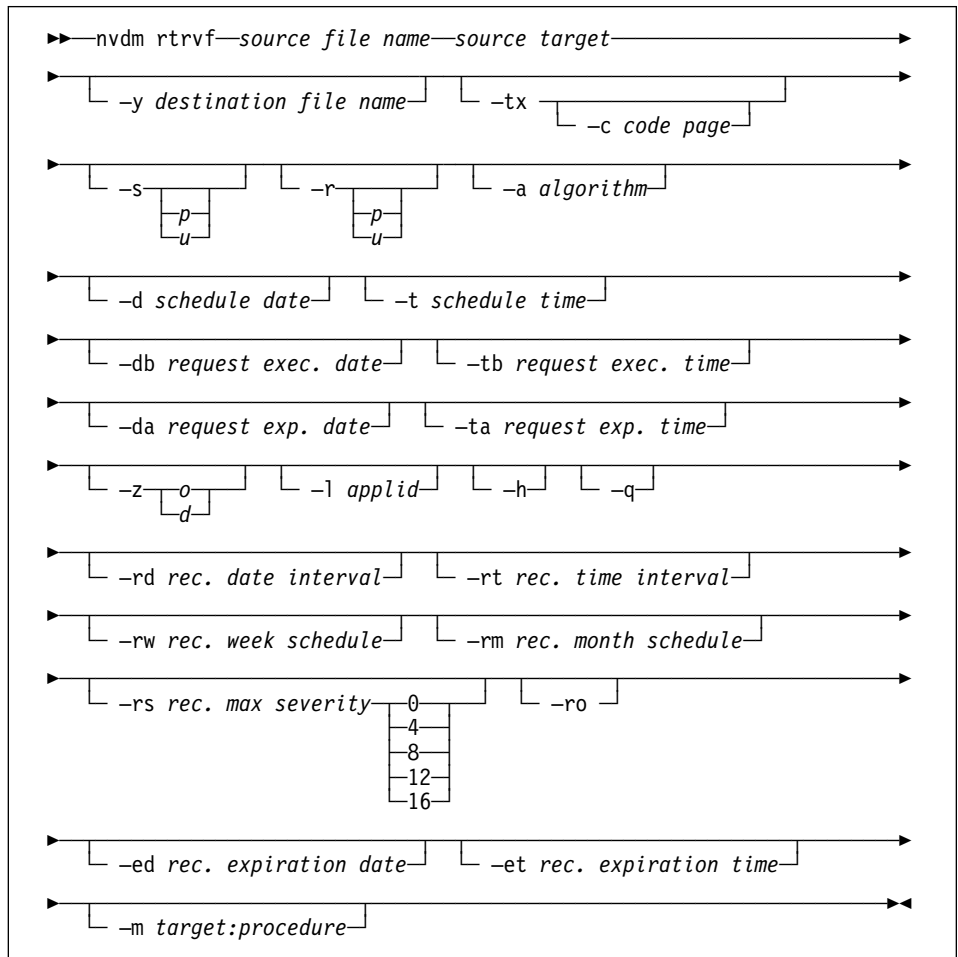
rtrvf – Retrieve a Noncataloged File

Use the **rtrvf** command to retrieve a file from a target where it is not cataloged, and store it on your target. Identify the file to retrieve using its local file name. If you do not specify a destination file name, the file is stored using its original file name.

This operation can only be performed between targets connected by the server-to-server (STS) transmission protocol (see the *TME 10 Software Distribution Quick Beginnings* manual).

This command cannot be used at mobile clients when working with the local catalog.

Syntax



Parameters

source file name

The local name of the file to be retrieved. This parameter is required.

source target

The target where the file to be retrieved is to be found.

-y *destination file name*

The local name of the file at the destination target. If you do not specify a destination file name, the following criteria apply when establishing the name of the file at destination:

- If the source file name you specify contains an absolute path (meaning that the name begins with the / character), and that path exists at the destination target, the file is stored in it. For example, if the source file name is /u/home/eurodocs/filename.txt, it will be stored in exactly the same way at its destination.
- If the source file name you specify is relative (meaning that the name does not begin with the / character), the file is stored in the local working directory at the destination target. The local working directory is defined in the PATHNAME field of the target's base configuration file (nvdn.cfg). If this PATHNAME is not specified, the default directory is /usr/lpp/netviewdm.

For example, if the source file name you specify is filename.txt, you do not specify a destination name, and PATHNAME is not specified in the configuration file, then the file will be stored as /usr/lpp/netviewdm/filename.txt at the destination target.

-tx The file to be retrieved is a text file. If you do not specify this parameter, the file is assumed to be binary.

-c *code page*

The code page of the text file. If you do not specify this parameter, the local system code page is used.

-s p|u

Transmit the file in packed (compressed) or unpacked (decompressed) format.

-sp File is to be packed

-su File is to be unpacked

If **-s** is omitted, no packing or unpacking is performed before transmitting the file.

-r p|u

Store the file in packed (compressed) or unpacked (decompressed) format.

-rp File is to be packed

-ru File is to be unpacked

If **-r** is omitted, no packing or unpacking is performed before storing the file.

-a *algorithm*

The algorithm to be used when compressing the file. Specify one of the following:

- SNA
- LZW
- File name of a user-supplied algorithm

If you omit this parameter, LZW is used.

-d *schedule date*

The date the request is to take place at the origin target.

Date and time parameters are described in detail in “Entering Dates and Times” on page 13.

-t *schedule time*

The time the request is to take place at the origin target.

-db *request execution date*

The date the request is to take place at the destination target.

-tb *request execution time*

The time the request is to take place at the destination target.

-da *request expiration date*

The date after which the request cannot take place at the destination target.

-ta *request expiration time*

The time after which the request cannot take place at the destination target.

-z *old*

The format of the execution time. Enter one of the following values:

- o** The time is expressed as the local time at the origin target. This is the default.
- d** The time is expressed as the local time at the destination target.

-l *application id*

The ID of the application used to submit the request. This ID can be used later to filter the requests submitted by a specific application.

It does not make sense to set an application ID for a request issued from the graphical interface, as this is an interactive process. Requests submitted from the graphical interface are automatically given the value of GUI as the application ID. You cannot change this value.

If you do not specify an application ID for requests issued from the command-line interface, the default value is CLI.

-h The request is to be submitted with high priority. You cannot specify this parameter if schedule times and execution times are specified.**-q** The request is to be submitted in held state. The request is scheduled only when released by the operator.

This parameter cannot be used locally by mobile clients.

-rd *rec. date interval*

A recursion date interval expressed in years, months, days. The request is executed upon every lapse of this interval. To specify an interval in terms of days, specify also the months and the years. To specify an interval in terms of months, specify also the years.

For example, 1 means the request is executed every year; 0,1 means the request is executed every month; 0,2,1 means the request is executed every day of every other month.

-rt *rec. time interval*

A recursion time interval expressed in hours and minutes. The request is executed upon every lapse of this interval. The format of a time interval is the same as for a time. For example, 3:45 means a time interval of 3 hours and 45 minutes.

-rw *rec. week schedule*

The days of the week when the recursive request is to take place. The days of the week are as follows:

0 = Sunday
 1 = Monday
 2 = Tuesday
 3 = Wednesday
 4 = Thursday
 5 = Friday
 6 = Saturday

For example, to set a recursion for each Monday and Friday of the week, enter:

```
-rw 1,5
```

-rm *rec. month schedule*

The days of the month when the recursive request is to take place. Each day of the month is represented by a number from 1 to 31.

For example, to select a recursion for the 15th and 27th of the month, enter:

```
-rm 15,27
```

The number 31 specifies the last day of any month. The number 29 specifies the last day of February even in a non-leap year.

-rs *rec. max severity 0|4|8|12|16*

A new occurrence of the request is generated only if the severity level returned from the previous occurrence is less than or equal to the max-severity specified.

Specify one of the following:

0 Successful
4 Warning
8 Error
12 Failed
16 Unrecoverable

rtrvf

- ro** Specify this parameter to stop the recursion mechanism if the current occurrence of the request is in progress when the new occurrence should be generated. If you do not specify this parameter, one or more occurrences may be skipped until the current one is completed, but the recursion mechanism is not stopped. This parameter is not set by default.
- ed** *rec. expiration date*
The date after which the system stops the recursion mechanism.
- et** *rec. expiration time*
The time after which the system stops the recursion mechanism.
- m** *target:procedure*
A cataloged procedure to execute on a given target when the request is completed. See “Running a Procedure after a Request Has Completed” on page 16.

Examples

```
nvdm rtrvf FILE1.TXT target1 -y FILE2.DAT -tx -c IBM-eucJP -sp -ru
```

This command retrieves FILE1.TXT from target1 and stores it a target FILE2.DAT. The file is a text file, the code page used is IBM-eucJP. The file is transmitted in a packed format and stored in unpacked format.

Related Information

The delf and the sendf command.

send – Send a File to Another Target

Use the **send** command to send a file, identified by its global name, to another target. If the file is not yet cataloged on the destination target, it is cataloged on arrival. Otherwise, it is stored under the local file name specified in the catalog entry at the destination. The file is not accepted if an existing catalog entry points to an existing local file.

You can transmit and store files in compressed or decompressed format across SNA or TCP/IP network links. You cannot change the compression format when transmitting within your domain, because the source and destination workstations share the same catalog and the compression format of the file must remain the same on each.

You can choose whether the transmitted file is to be compressed or decompressed at local server or destination target. Set the ACT_ON_TARGET environment variable by entering the following command at a command line:

```
set ACT_ON_TARGET=0 | 1
```

where 0 and 1 have the following meaning:

- 0** When a remote server sends a file in compress format and wants to store it in decompress format at destination target, the file is decompressed at local server and sent in decompress format to destination target.

When a remote server retrieves a file in compress format the file is compressed at local server and sent in compress format to the destination server.

- 1** When a remote server sends a file in compress format and wants to store it in decompress format at destination target, the file is sent in compress format and it is decompressed at destination target.

When a remote server retrieves a file in compress format, the file is compressed at source target and transmitted as compressed.

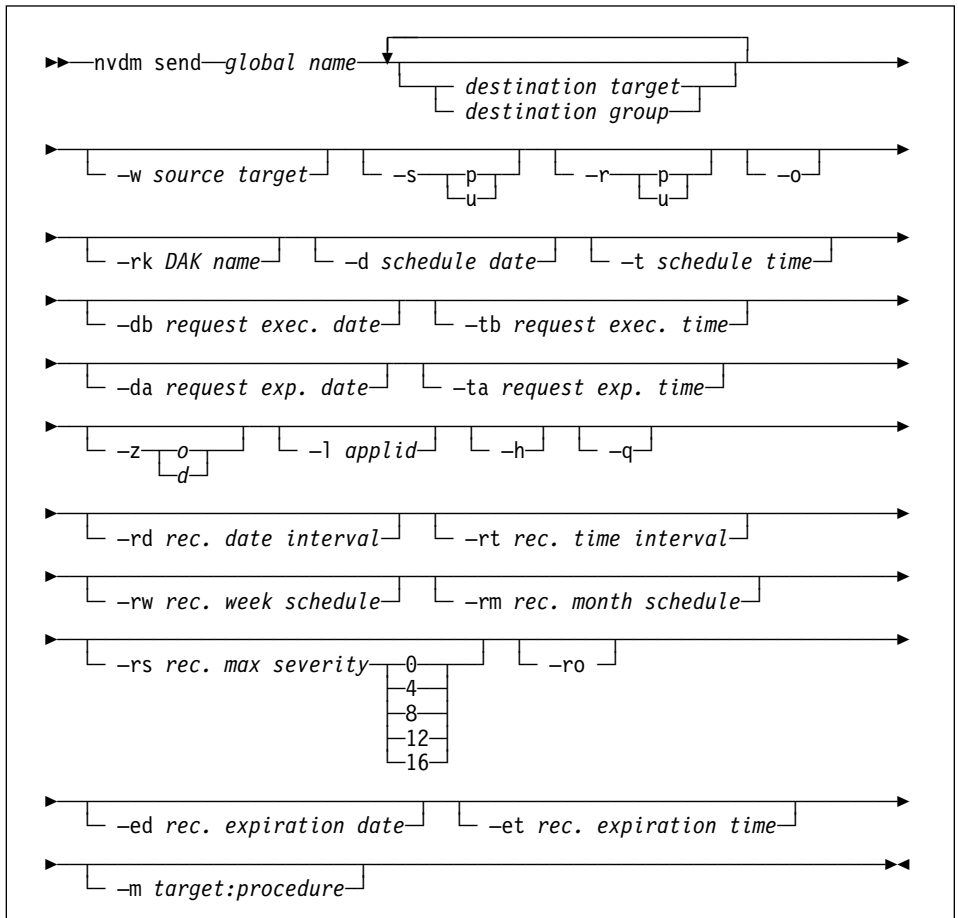
For targets that have installed TME 10 Software Distribution versions prior to 3.1, this variable can be set only to 0. The default is 0.

When you have more than one focal point defined, you must specify the name of the destination target.

This command cannot be used at mobile clients when working with the local catalog.

send

Syntax



Parameters

global name

The global name of the file. See “Entering Global Names” on page 13.

You can use wildcard characters, but the function is scheduled only if a unique match is found.

destination target | destination group

The target or group to which the file is to be sent. You can enter up to 16 destination names. If it is not specified, the file is sent to the network’s focal point.

-w *source target*

The target on which the file to be sent is to be found (must be the local target or the server). See “Entering Target Names” on page 12. If no **-w** parameter is specified, the file is assumed to be stored on your computer.

-s p|u (*sender's action*)

Transmit the file in packed (compressed) or unpacked (decompressed) format.

-sp File is to be packed

-su File is to be unpacked

If **-s** is omitted, no packing or unpacking is done before sending the file.

-r p|u (*receiver's action*)

Store the file in packed (compressed) or unpacked (decompressed) format after it has been received at the destination.

-rp File is to be packed

-ru File is to be unpacked

If **-r** is omitted, no packing or unpacking is done after receiving the file.

This parameter cannot be specified for send operations to focal points running NetView DM.

- o** Overwrite the existing file. If this parameter is specified, the existing file with the same name is overwritten. Use this parameter only from a remote administration site.

-rk *DAK name*

The file is retrieved with the data access key specified. If **-rk** is not specified, the file is retrieved with the data access key specified in the catalog.

-d *schedule date*

The date the request is to take place at the origin target.

Date and time parameters are described in detail in “Entering Dates and Times” on page 13.

-t *schedule time*

The time the request is to take place at the origin target.

-db *request execution date*

The date the request is to take place at the destination target.

-tb *execution time*

The time the request is to take place at the destination target.

-da *request expiration date*

The date after which the request cannot take place at the destination target.

-ta *request expiration time*

The time after which the request cannot take place at the destination target.

-z o|d

The format of the execution time. Enter one of the following values:

- o** The time is expressed as the local time at the origin target. This is the default.
- d** The time is expressed as the local time at the destination target.

- h** The request is to be submitted with high priority. You cannot specify this parameter if schedule times and execution times are specified.
- q** The request is to be submitted in held state. The request is scheduled only when released by the operator.

This parameter cannot be used locally by mobile clients.

-l *application id*

The ID of the application used to submit the request. This ID can be used later to filter the requests submitted by a specific application.

It does not make sense to set an application ID for a request issued from the graphical interface, as this is an interactive process. Requests submitted from the graphical interface are automatically given the value of GUI as the application ID. You cannot change this value.

If you do not specify an application ID for requests issued from the command-line interface, the default value is CLI.

-rd *rec. date interval*

A recursion date interval expressed in years, months, days. The request is executed upon every lapse of this interval. To specify an interval in terms of days, specify also the months and the years. To specify an interval in terms of months, specify also the years.

For example, 1 means the request is executed every year; 0,1 means the request is executed every month; 0,2,1 means the request is executed every day of every other month.

-rt *rec. time interval*

A recursion time interval expressed in hours and minutes. The request is executed upon every lapse of this interval. The format of a time interval is the same as for a time. For example, 3:45 means a time interval of 3 hours and 45 minutes.

-rw *rec. week schedule*

The days of the week when the recursive request is to take place. The days of the week are as follows:

0 = Sunday
1 = Monday
2 = Tuesday
3 = Wednesday
4 = Thursday
5 = Friday
6 = Saturday

For example, to set a recursion for each Monday and Friday of the week, enter:

-rw 1,5

-rm *rec. month schedule*

The days of the month when the recursive request is to take place. Each day of the month is represented by a number from 1 to 31.

For example, to select a recursion for the 15th and 27th of the month, enter:

```
-rm 15,27
```

The number 31 specifies the last day of any month. The number 29 specifies the last day of February even in a non-leap year.

-rs *rec. max severity* **0|4|8|12|16**

A new occurrence of the request is generated only if the severity level returned from the previous occurrence is less than or equal to the max-severity specified.

Specify one of the following:

- 0** Successful
- 4** Warning
- 8** Error
- 12** Failed
- 16** Unrecoverable

-ro Specify this parameter to stop the recursion mechanism if the current occurrence of the request is in progress when the new occurrence should be generated. If you do not specify this parameter, one or more occurrences may be skipped until the current one is completed, but the recursion mechanism is not stopped. This parameter is not set by default.

-ed *rec. expiration date*

The date after which the system stops the recursion mechanism.

-et *rec. expiration time*

The time after which the system stops the recursion mechanism.

-m *target:procedure*

A cataloged procedure to execute on a given target when the request is completed. See “Running a Procedure after a Request Has Completed” on page 16.

Examples

```
nvdmsend EURO.SPELLCHECK.EXE.1.US FREDWS
```

This command causes the file EURO.SPELLCHECK.EXE.1.US to be sent to the target FREDWS. The file is not compressed. The file is located on your workstation, because **-w** is not specified. The send takes place as soon as possible.

Related Information

The rtrv, del, and cat commands.

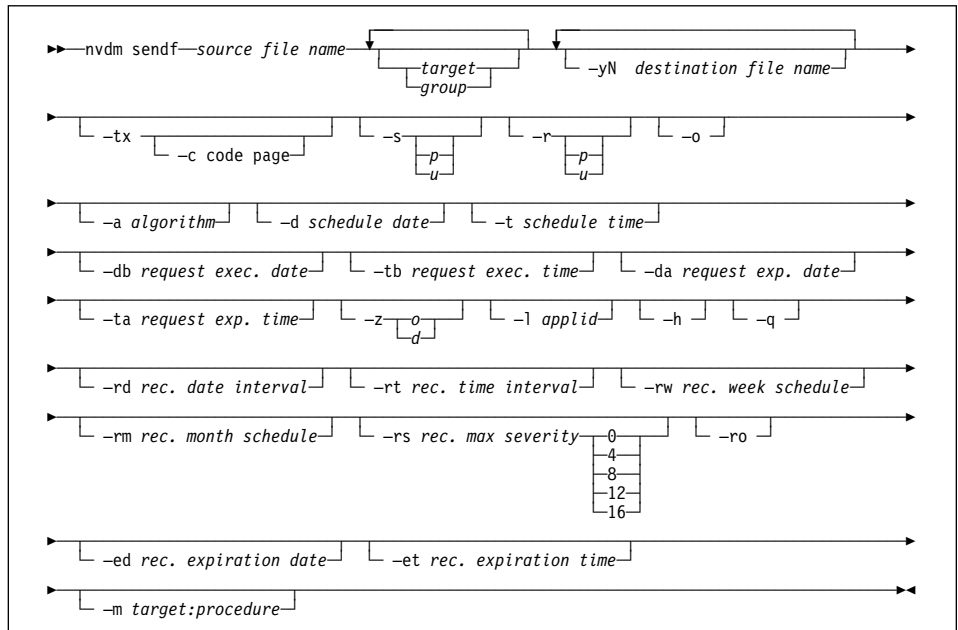
sendf – Send a Noncataloged File

Use the **sendf** command to send a noncataloged file from your target to one or more TME 10 Software Distribution targets. Identify the file to be sent by its local name. The file is stored at each destination target using the local destination name specified in the command. A different destination name can be specified for each target. If no destination name is specified, the file is stored locally using the same name it had at its origin target.

This operation can only be performed between targets connected by the server-to-server (STS) transmission protocol (see the *TME 10 Software Distribution Quick Beginnings* manual).

This command cannot be used at mobile clients when working with the local catalog.

Syntax



Parameters

source file name

The local name of the file to be sent. This parameter is required. You can use wildcards, but the command is performed only if a unique match is found.

target | group

The destination targets and groups where the file is to be sent.

-yN *destination file name*

Specifies additional local names of the file at each destination target, depending on target's file system. **N** indicates the specific file system ID the local file name refers to.

The file system ID associated with every workstation is defined in its base configuration file (nvdm.cfg). Enter a value from 1 to 7 for **N**. You can specify up to seven file system IDs (together with the local name) for each file that you are cataloging.

Each platform picks the correct local name on the basis of its file system ID. For example, if you define file system ID=5 in the base configuration file of all OS/2 workstations, when you catalog a file to be sent to these workstations you have to associate the value 5 with the OS/2 local name.

If you do not define the file system ID in the base configuration file of the OS/2 workstations, when you catalog a file for OS/2 you must associate the default file system ID value, (which for OS/2 is 2), with the OS/2 local name. With this value the file that you send is stored in the c:\user\lpp\netviewdm\repos directory. The following table lists the file system ID default values for the various platforms:

Table 8. Default File System IDs

Platform	File System ID	File Name Syntax
AIX/6000	1	/usr/lpp/netviewdm/repos...
DOS, OS/2, Windows 3.11	2	c:\softdist\repos...
NetWare	3	sys:system\ibmnvdm\repos...
Windows 95, Windows 98, Windows 2000, Windows NT	5	c:\softdist\repos...

If you do not specify the file system ID value for the OS/2 workstations, the default local name follows the syntax rules associated with file system ID = 0. If you do not define the local name associated with 0, the default local name follows the syntax of the source workstation.

You can catalog a change file specifying different file system IDs, but all change control operations use the value 0.

-tx The file to be sent is a text file. If you do not specify this parameter, the file is assumed to be binary.

-c *code page*

The code page of the text file. It must be specified using the long format. If you do not specify this parameter, the local system code page is used.

-s plu

Send the file in packed (compressed) or unpacked (decompressed) format.

-sp File is to be packed

-su File is to be unpacked

If **-s** is omitted, no packing or unpacking is performed before sending the file.

-r p|u

Store the file in packed (compressed) or unpacked (decompressed) format.

-rp File is to be packed

-ru File is to be unpacked

If **-r** is omitted, no packing or unpacking is performed before storing the file.

- o** Overwrite the existing file. If this parameter is specified, the existing file with the same name is overwritten. Use this parameter only from a remote administration site.

-a *algorithm*

The algorithm to be used when compressing the file. Specify one of the following:

- SNA
- LZW
- Filename of a user supplied algorithm

If you omit this parameter, LZW is used.

-d *schedule date*

The date the request is to take place at the origin target.

Date and time parameters are described in detail in “Entering Dates and Times” on page 13.

-t *schedule time*

The time the request is to take place at the origin target.

-db *request execution date*

The date the request is to take place at the destination target.

-tb *request execution time*

The time the request is to take place at the destination target.

-da *request expiration date*

The date after which the request cannot take place at the destination target.

-ta *request expiration time*

The time after which the request cannot take place at the destination target.

-z o|d

The format of the execution time. Enter one of the following values:

- o** The time is expressed as the local time at the origin target. This is the default.
- d** The time is expressed as the local time at the destination target.

-l *application id*

The ID of the application used to submit the request. This ID can be used later to filter the requests submitted by a specific application.

It does not make sense to set an application ID for a request issued from the graphical interface, as this is an interactive process. Requests submitted from the

graphical interface are automatically given the value of GUI as the application ID. You cannot change this value.

If you do not specify an application ID for requests issued from the command-line interface, the default value is CLI.

- h The request is to be submitted with high priority. You cannot specify this parameter if schedule times and execution times are specified.
- q The request is to be submitted in held state. The request is scheduled only when released by the operator.

This parameter cannot be used locally by mobile clients.

-rd *rec. date interval*

A recursion date interval expressed in years, months, days. The request is executed upon every lapse of this interval. To specify an interval in terms of days, specify also the months and the years. To specify an interval in terms of months, specify also the years.

For example, 1 means the request is executed every year; 0,1 means the request is executed every month; 0,2,1 means the request is executed every day of every other month.

-rt *rec. time interval*

A recursion time interval expressed in hours and minutes. The request is executed upon every lapse of this interval. The format of a time interval is the same as for a time. For example, 3:45 means a time interval of 3 hours and 45 minutes.

-rw *rec. week schedule*

The days of the week when the recursive request is to take place. The days of the week are as follows:

- 0 = Sunday
- 1 = Monday
- 2 = Tuesday
- 3 = Wednesday
- 4 = Thursday
- 5 = Friday
- 6 = Saturday

For example, to set a recursion for each Monday and Friday of the week, enter:

```
-rw 1,5
```

-rm *rec. month schedule*

The days of the month when the recursive request is to take place. Each day of the month is represented by a number from 1 to 31.

For example, to select a recursion for the 15th and 27th of the month, enter:

```
-rm 15,27
```

The number 31 specifies the last day of any month. The number 29 specifies the last day of February even in a non-leap year.

sendf

-rs *rec. max severity* **0|4|8|12|16**

A new occurrence of the request is generated only if the severity level returned from the previous occurrence is less than or equal to the max-severity specified.

Specify one of the following:

- 0** Successful
- 4** Warning
- 8** Error
- 12** Failed
- 16** Unrecoverable

-ro Specify this parameter to stop the recursion mechanism if the current occurrence of the request is in progress when the new occurrence should be generated. If you do not specify this parameter, one or more occurrences may be skipped until the current one is completed, but the recursion mechanism is not stopped. This parameter is not set by default.

-ed *rec. expiration date*

The date after which the system stops the recursion mechanism.

-et *rec. expiration time*

The time after which the system stops the recursion mechanism.

-m *target:procedure*

A cataloged procedure to execute on a given target when the request is completed. See “Running a Procedure after a Request Has Completed” on page 16.

Examples

```
nvdmsendf FILE1.TXT target2 target3 group1  
-t -y0 dest_file_name1 -y1 dest_file_name2 -sp -ru
```

This command causes the text file FILE1.TXT located at your target to be sent, in packed format, to target2, target3, and to all the targets belonging to group1; the file is stored in unpacked format at the destination targets with dest_file_name1, if the target file system is 0, or with dest_file_name2, if the target file system is 1.

Related Information

The delf and the rtrvf commands.

start – Start TME 10 Software Distribution on Your Workstation

Use the **start** command to start TME 10 Software Distribution on your workstation. After installation, by default, TME 10 Software Distribution is started automatically during the workstation startup process. Use this command if you need to start the product manually.

The **start** command can be issued only:

- By a user logged on to the workstation as root
- By a user associated with the FNDADMN group

Syntax

▶▶—nvdm start—▶◀

Related Information

The stop and relc commands.

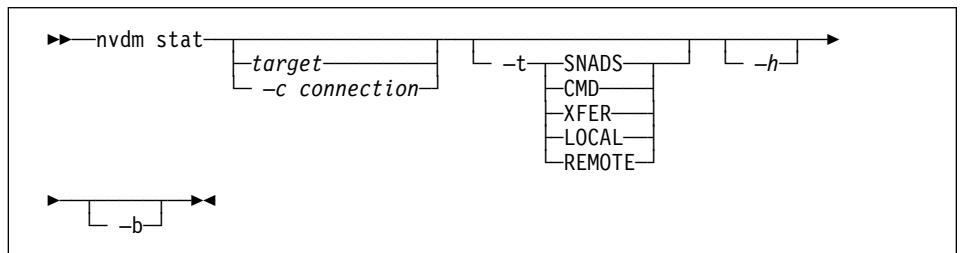
stat – Check the Status of a Communication Queue

Use the **stat** command to return the status of TME 10 Software Distribution communication, showing the following:

- The status of communication to and from remote targets either on SNA/DS connections or on server-to-server (STS) connections for commands or files transfer. Status can be:
 - Released
 - Held
- For each communication queue:
 - The queue name
 - The queue type
 - The connection associated (for remote queues only)
 - The number of contained entries
 - The queue status:
 - Held
 - Released
 - Error — Held in an error state
 - Resynch — Held waiting for resynchronization after an error
 - Waiting — Temporarily held
 - The target status:
 - Held
 - Released

This command cannot be used at mobile clients when working with the local catalog.

Syntax



Parameters

target

The name of the target that the queue refers to. If the queue is on a local target, this parameter is the queue name; if the queue is on a remote target, this parameter is the name of the next server connected.

You can use wildcard characters.

This parameter is optional. If this or the **-c** parameter is not specified, the default value is all remote targets.

-c connection

The name of the connection the queue refers to. This is the name of the file where the connection is specified, in the /usr/lpp/netviewdm/db/snadscon directory.

This parameter applies only to queues on remote targets, because local queues are not associated with any connections.

You can use wildcard characters.

-t SNADS|CMD|XFER|LOCAL|REMOTE

The type of queue. It can be one of the following:

- SNADS** On SNA/DS connections
- CMD** On server-to-server (STS) connections to transfer commands
- XFER** On server-to-server (STS) connections to transfer files
- LOCAL** Local
- REMOTE** Remote

This parameter is optional. If you do not specify it, the status of all the remote queues is displayed.

- h** Display only the queues whose status is held. This parameter is optional.
- b** Display only the busy queues (that is, queues containing at least one entry). This parameter is optional.

Output

The output of the **stat** command is as follows:

```
SNADS: [Released | Held | Not available ]
XFER : [Released | Held | Not available ]
CMD  : [Released | Held | Not available ]
```

Target	Type	Connection	Entries	Q Status	Tg Status
[tgt1]	[type1]	[conn1]	[n]	[status1]	
[tgt2]	[type2]	[conn2]	[n]	[status2]	
:	:	:	:	:	
:	:	:	:	:	

Examples

```
nvdms stat target1
```

The output of this command looks like the following:

stat

SNADS: Not Available
XFER : Released
CMD : Released

Target	Type	Connection	Entries	Q Status	Tg Status
target1	local		1	Released	Released

Related Information

The hldc, relc, lsq, and prgq commands.

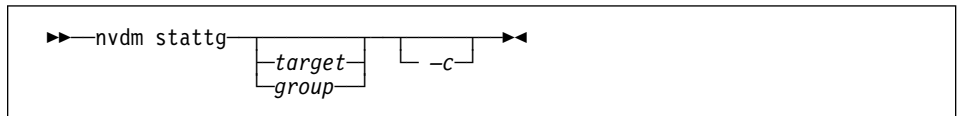
stattg – List Target Status

Use the **stattg** command to list the status of one or all local targets. Target status can be one of the following:

- Available; ready to process a request
- Available/held; ready to process a request but held
- Not available; not running or a connection with it cannot be established
- Not available/held; not running or not reachable and held
- Busy; processing a request or not working and a request is scheduled for it
- Busy/held; processing a request or not working and a request is scheduled for it and also held
- Unknown; agent was polled using different workstation name than that defined locally in the client configuration file (only available if **-c** parameter supplied).

You must have at least View or Modify System Administration authorization to use this command.

Syntax



Parameters

target | group

The local target or group whose status is to be listed.

You can use an asterisk (*) enclosed in single or double quotation marks to list the status of all targets. If you do so, a message is displayed warning you that running the command could take a long time; you can either confirm or cancel the command.

This parameter is optional.

-c Report with status *Unknown* if the hostname at the target workstation has a different workstation name defined in the client configuration file than that supplied by the host.

This parameter to the **stattg** command can be used to identify workstations where the server has more than one workstation name in its database with the same hostname.

For example, if a workstation has been re-defined to the server for some reason, the operator may have supplied a different workstation name than that originally used, but have used the original hostname. In this event, the agent now has the new workstation name, but the server has both workstation names defined, and if the **-c** parameter is not used the agent reports itself as being *Available* under both

workstation names. By using the `-c` parameter, polling both workstation names will allow you to identify which is the correct one, as one will return the status *Available* and the other *Unknown*.

However, before using this parameter you should consider the question of the timing of the `stattg` requests. When an agent receives a `stattg` request it sends the status to the server but is then not immediately available to satisfy another request. This means that a second request, received within, say, one minute of the first request, will return the status *Not Available*. If you are polling two suspect workstation names you should wait for this period before sending the second request.

This also means that if you send a `stattg` request using the asterisk wildcard to obtain the status of all or a group of workstations, the results received will depend on whether the *incorrect* workstation name comes before or after the *correct* one in the server's database:

Incorrect workstation name is polled first

The status of the *incorrect* workstation name will be given as *Unknown*, while the *correct* workstation will give *Not Available*

Correct workstation name is polled first

The status of the *correct* workstation name will be given as *Available* while the *incorrect* workstation will give *Not Available*

Thus, after using the asterisk wildcard with the `-c` parameter, you should individually poll each workstation name given as *Not Available*, waiting for approximately one minute before issuing each command. Workstations that are genuinely unavailable will report the same status as before; workstations that were unavailable while they were recovering from a previous `stattg` command will now report their true status.

This parameter is optional.

Examples

The output of the **stattg** command is as follows:

Target	Status
[ws_id]	[status]

The status can be one of the following:

- Available
- Available, held
- Not available
- Not available, Held
- Busy
- Busy, held
- Unknown

Examples

| nvdm stattg '*' -c

This command lists the connection status for all local targets. The output looks like the following:

Target	Status
WS01	Busy
WS02	Not Available
WS03	Available, Held
WS04	Unknown

Related Information

The hldtg and reltg commands.

stop

stop – Stop TME 10 Software Distribution on Your Workstation

Use the **stop** command to stop TME 10 Software Distribution on the workstation at which the command is executed. A parameter specifies which of the two types of stop is performed.

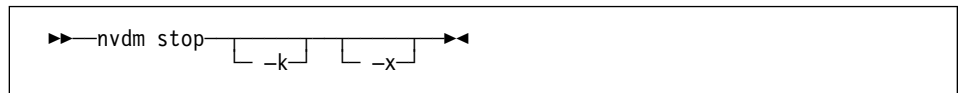
Graceful All change control tasks that are currently executing on the server are completed, but no new ones are started. When all current activity is complete, TME 10 Software Distribution stops.

Immediate All activity is stopped, after the tasks currently executing on the server are completed.

The **stop** command can be issued only:

- By a user logged on to the workstation as root
- By a user associated with the FNDADMN group

Syntax



Parameters

-k TME 10 Software Distribution tasks are terminated, but not transmission processes. If **-k** is omitted, a graceful termination is performed.

If the **-x** parameter is not specified, clients still have access to the server, although no change control or distribution activities can take place.

-x Shut down both the base and server options. This parameter only has effect on a base system with the server option installed. All transmission tasks are terminated, but only after completing the operations that are currently executing on the server.

Examples

```
nvdm stop -k -x
```

This command terminates all the tasks, after completing those that are currently executing on the server.

Related Information

The start and hldc commands.

svr – Select a Server

Use the **svr** command to open a connection to a TME 10 Software Distribution server.

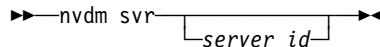
The first time TME 10 Software Distribution is invoked after its installation, if no **svr** command is issued before another command, the system scans through the list of servers specified in the target configuration and connects to the first server on which the target is configured. For subsequent startups or after a reinstallation, if no **svr** command is issued, the system will reconnect to the last server it was connected to during previous operations.

Mobile clients use this command to connect to their local catalog by specifying the server name MYSELF, as follows:

```
nvdm svr myself
```

Note that no server in the network can be named MYSELF.

Syntax



```
nvdm svr [server id]
```

Parameters

server id

The name of the server to connect to. If this parameter is not specified, a connection is made to the default server (the one used by your target for change control and distribution requests).

To work locally at a mobile client, specify `myself` as the server ID.

Examples

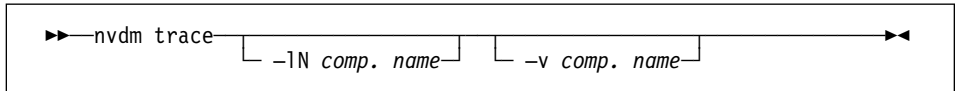
```
nvdm svr SERVER02
```

This command specifies that from now on all commands are sent to the server SERVER02. Commands will continue to be sent there until another **svr** command is issued.

trace – Set Internal Tracing

Use the **trace** command to turn internal tracing on or off, or to change the level of tracing being performed for major product components. You can issue this command while the product is running; you do not have to issue `nvdms stop` and `nvdms start` commands to do so.

Syntax



Parameters

-IN *component name*

Sets the level of tracing to apply and the component to which it applies. **N** is a number from 0 to 6 that indicates the level of tracing, as follows:

- 0 All functions flow is traced. It is the highest level of tracing.
- 1 Internal function flow is traced.
- 2 Major function flow and interactions are traced.
- 3 Major paths and special conditions are traced.
- 4 Special conditions are traced.
- 5 Major errors and termination errors are traced.
- 6 No tracing is performed.

Specify the name of the component to which the tracing applies, as described in Table 9 on page 199.

Table 9. Values for Component Names

Value	Component	Value	Component
cc	common change file functions library	cl	command line interface
cm	change manager (DACA)	co	common routines library
db	database access services library	fs	file services library
gi	graphic user interface	ngin	network gateway in (outgoing requests)
nglib	network gateway library	ngout	network gateway out (incoming requests)
ngprs	network gateway parser	pl	plan library
rb	request block handler (RB-API)	rhlib	request handler library
rhmn	request handler main task	rhpt	request handler processing task
rte	RB-API router	rs	resource status library
rx	request transport (RB-API)	sa	scheduler administration library
sf	shared functions library	sh	scheduler
sts	server-to-server	sw	scheduled work library
tc	SNA/DS transmission controller	tclib	Transmission controller library
tr	TC Receive task	ts	TC Send task
uc	User interface common functions library	vl	Validation library
base	All but the ones listed above	all	All libraries and executables

If you specify **all** as the the component name, its level overrides the levels set for any other components.

You can enter this parameter up to 16 times.

-v component name

Views the current level of tracing you set. Specify the name of the component to which the tracing applies to, (see Table 9).

You can enter this parameter up to 16 times.

Examples

```
nvdm trace -l2 co -l3 db
```

This command sets the level of tracing for common routines to 2 and the level of tracing for DAS to 3.

Related Information

The `tron` and `troff` command.

troff – Stop a Trace

Use the **troff** command to stop TME 10 Software Distribution from producing trace output for communication between the target where the command is submitted and the server.

Syntax

```
▶▶—nvdm troff—◀◀
```

Examples

```
nvdm troff
```

This command stops the server from producing trace output for communication with your computer.

Related Information

The tron command.

tron – Start a Trace

Use the **tron** command to cause TME 10 Software Distribution to produce trace output for communication between the target where the command is submitted and the server.

Syntax

```
▶▶—nvdm tron—◀◀
```

Examples

```
nvdm tron
```

This command makes the server produce trace output for communication with your computer. The output file (`/usr/lpp/netviewdm/fndapi`) is located at the server.

Related Information

The troff command.

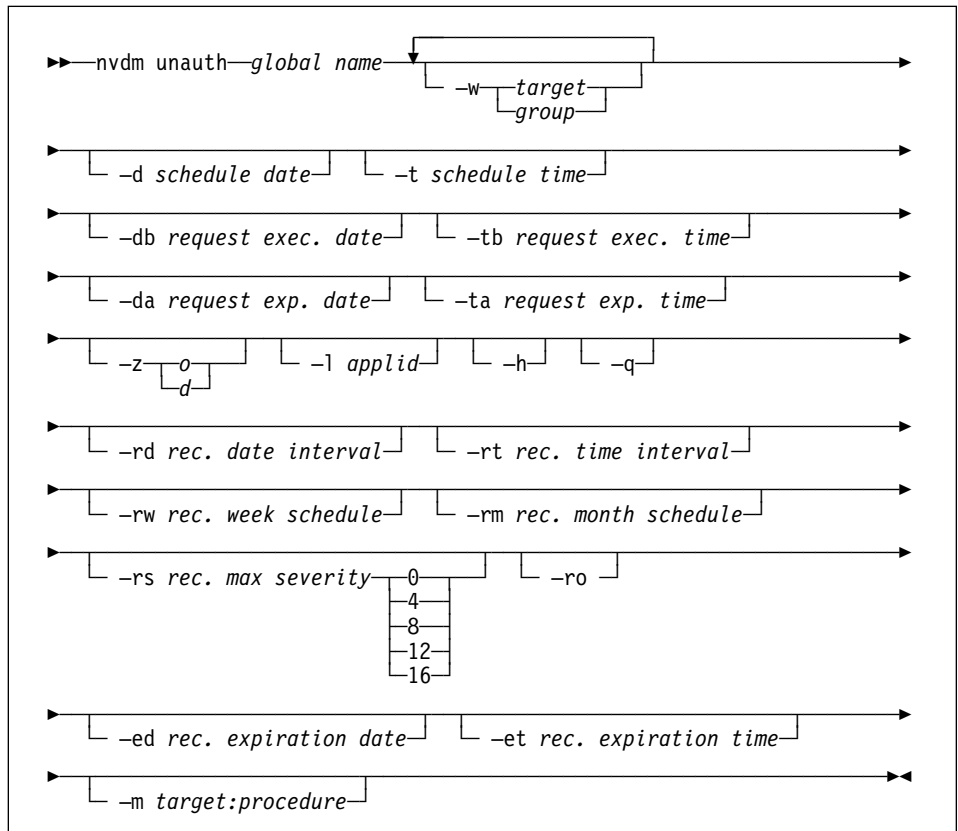
The section of the *User's Guide* about controlling messages and traces.

unauth – Remove a Target Authorization to Access a File

Use the **unauth** command to remove authorization of a catalog entry on a target or group. This action prevents those targets from viewing or using the catalog entry.

This operation can be performed on remote targets only if the targets involved in the operation are connected by the server-to-server (STS) transmission protocol (see the *TME 10 Software Distribution Quick Beginnings* manual).

Syntax



Parameters

global name

Global name of the catalog entry to unauthorize. See “Entering Global Names” on page 13.

You can use wildcard characters, but a unique match must be found.

-w *target | group*

The targets or groups to which this command applies. See “Entering Target Names” on page 12.

-d *schedule date*

The date the request is to take place at the origin target.

Date and time parameters are described in more detail in “Entering Dates and Times” on page 13.

-t *schedule time*

The time the request is to take place at the origin target.

-db *request execution date*

The date the request is to take place at the destination target.

-tb *request execution time*

The time the request is to take place at the destination target.

-da *request expiration date*

The date after which the request cannot take place at the destination target.

-ta *request expiration time*

The time after which the request cannot take place at the destination target.

-z o|d

The format of the execution time. Enter one of the following values:

o The time is expressed as the local time at the origin target. This is the default.

d The time is expressed as the local time at the destination target.

-h The request is to be submitted with high priority. You cannot specify this parameter if schedule times and execution times are specified.

-q The request is to be submitted in held state. The request is scheduled only when released by the operator.

This parameter cannot be used locally by mobile clients.

-l *application id*

The ID of the application used to submit the request. This ID can be used later to filter the requests submitted by a specific application.

It does not make sense to set an application ID for a request issued from the graphical interface, as this is an interactive process. Requests submitted from the graphical interface are automatically given the value of GUI as the application ID. You cannot change this value.

If you do not specify an application ID for requests issued from the command-line interface, the default value is CLI.

-rd *rec. date interval*

A recursion date interval expressed in years, months, days. The request is executed upon every lapse of this interval. To specify an interval in terms of days, specify also the months and the years. To specify an interval in terms of months, specify also the years.

For example, 1 means the request is executed every year; 0,1 means the request is executed every month; 0,2,1 means the request is executed every day of every other month.

-rt *rec. time interval*

A recursion time interval expressed in hours and minutes. The request is executed upon every lapse of this interval. The format of a time interval is the same as for a time. For example, 3:45 means a time interval of 3 hours and 45 minutes.

-rw *rec. week schedule*

The days of the week when the recursive request is to take place. The days of the week are as follows:

- 0 = Sunday
- 1 = Monday
- 2 = Tuesday
- 3 = Wednesday
- 4 = Thursday
- 5 = Friday
- 6 = Saturday

For example, to set a recursion for each Monday and Friday of the week, enter:

```
-rw 1,5
```

-rm *rec. month schedule*

The days of the month when the recursive request is to take place. Each day of the month is represented by a number from 1 to 31.

For example, to select a recursion for the 15th and 27th of the month, enter:

```
-rm 15,27
```

The number 31 specifies the last day of any month. The number 29 specifies the last day of February even in a non-leap year.

-rs *rec. max severity* **0|4|8|12|16**

A new occurrence of the request is generated only if the severity level returned from the previous occurrence is less than or equal to the max-severity specified.

Specify one of the following:

- 0** Successful
- 4** Warning
- 8** Error
- 12** Failed
- 16** Unrecoverable

-ro Specify this parameter to stop the recursion mechanism if the current occurrence of the request is in progress when the new occurrence should be generated. If you do not specify this parameter, one or more occurrences may be skipped until the current one is completed, but the recursion mechanism is not stopped. This parameter is not set by default.

-ed *rec. expiration date*

The date after which the system stops the recursion mechanism.

-et *rec. expiration time*

The time after which the system stops the recursion mechanism.

-m *target;procedure*

A cataloged procedure to execute on a given target when the request is completed.

See “Running a Procedure after a Request Has Completed” on page 16.

Examples

```
nvdms unauth EURO.WORDPROC.* -w FREDWS
```

This command removes the authorization for FREDWS to view or use any of the catalog entries relating to the Eurotravel word processor.

Related Information

The auth and ls commands.

unbld – Unpack the components of a change file

Use the **unbld** command to unpack the component files that make up a change file. The files are restored on their source path and can be used to build a new change file.

The original change file still exists and is cataloged. To build the same change file again, you must uncatalog it.

If the path of a component file is not specified in the change file profile, the file is restored under the current directory. Only the files actually included in the change file are unpacked; any remote files accessed by the change file are not restored. The **unbld** command is not effective for objects with their TYPE set to:

- REMOTE_FILE
- REMOTE_FILE_WITH_TOKENS
- DIRECTORY
- REMOTE_IMAGE
- REMOTE_IMAGE_PMP

The **unbld** command is not effective for objects with their ACTION set to:

- CREATE
- DELETE

For these objects, the command does not take any action, and only valid objects are unpacked.

Use the **unbld** command only with generic and installp change files. Only a user defined as root to the operating system can unbld installp change files.

The component files are restored with the following attributes:

Owner The user who submitted the **unbld** command.

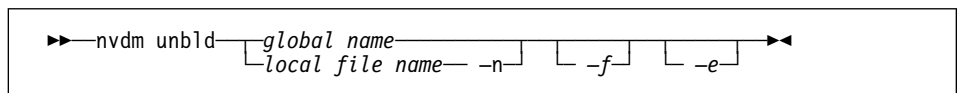
Group FNDADMN

UNIX attributes

The user who entered the command has read, write, and execute permission. Users belonging to the FNDADMN group have read and execute permission.

This command cannot be used at mobile clients when working with the local catalog.

Syntax



Parameters

global name

The global name of the change file to be unbld. You can use wildcard characters, but a unique match must be found.

local file name -n

The local name of the change file on your machine. Specify the **-n** parameter to differentiate this name from the global name. You can use wildcard characters, but a unique match must be found.

- f** Forces an overwrite if you try to unbld a change file and the component files already exist on your machine. If this option is not specified and a local file already exists, you are warned and the command is not executed.
- e** Extend file systems. If this parameter is specified, the journaled file system is extended if additional disk space is required when storing the component files.

Examples

```
nvdm unbld CHANGE.FILE.REF.1 -f
```

This command unpacks the component files that make up the change file cataloged as CHANGE.FILE.REF.1. Suppose the change file is associated with the following profile:

```
GLOBAL NAME:          CHANGE.FILE.REF.1
CHANGE FILE TYPE:     GEN

OBJECT:
  SOURCE NAME:         /usr/lpp/file.a
  TARGET NAME:         /usr/file.a
  TYPE:                FILE
  ACTION:              COPY

OBJECT:
  SOURCE NAME:         /usr/lpp/file.b
  TARGET NAME:         /usr/file.b
  TYPE:                FILE
  ACTION:              COPY

OBJECT:
  SOURCE NAME:         file.c
  TARGET NAME:         /usr/file.c
  TYPE:                FILE
  ACTION:              COPY

OBJECT:
  SOURCE NAME:         /usr/lpp/file.d
  SOURCE AT INSTALL:   /mount/usr/lpp/file.d
  TARGET NAME:         /usr/file.d
  TYPE:                REMOTE FILE
  ACTION:              COPY
```

unbld

The **unbld** command unpacks file.a and file.b in /usr/lpp and file.c under the current directory. These files are unpacked replacing the existing ones (**-f** option). The command does not take any action on file.d, because this is a remote file.

You can use the unpacked files to create a new change file or you can uncatalog the original change file, modify its profile, and build the same change file again.

Related Information

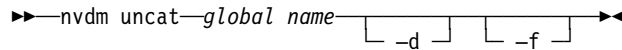
The lscf and bld commands.

uncat – Remove a Catalog Entry

Use the **uncat** command to remove a catalog entry. There is an option to delete the local file referred to in the catalog entry at the same time.

This command can be used locally by mobile clients.

Syntax



Parameters

global name

Global name of the catalog entry. See “Entering Global Names” on page 13.

You can use wildcard characters, but the function is scheduled only if a unique match is found.

- d** Deletes the local file referred to in the catalog entry.
- f** Force the uncatalog to take place without prompting for you to confirm it. If **-f** is not specified, you are asked to confirm the uncatalog by typing y.

Examples

nvdms uncat EURO.SPELLCHECK.DAT.3.US

This command removes the catalog entry `EURO.SPELLCHECK.DAT.3.US` from the catalog, but does not delete the associated local file from the server.

Related Information

The cat command.

uninst – Remove a Component from a Target

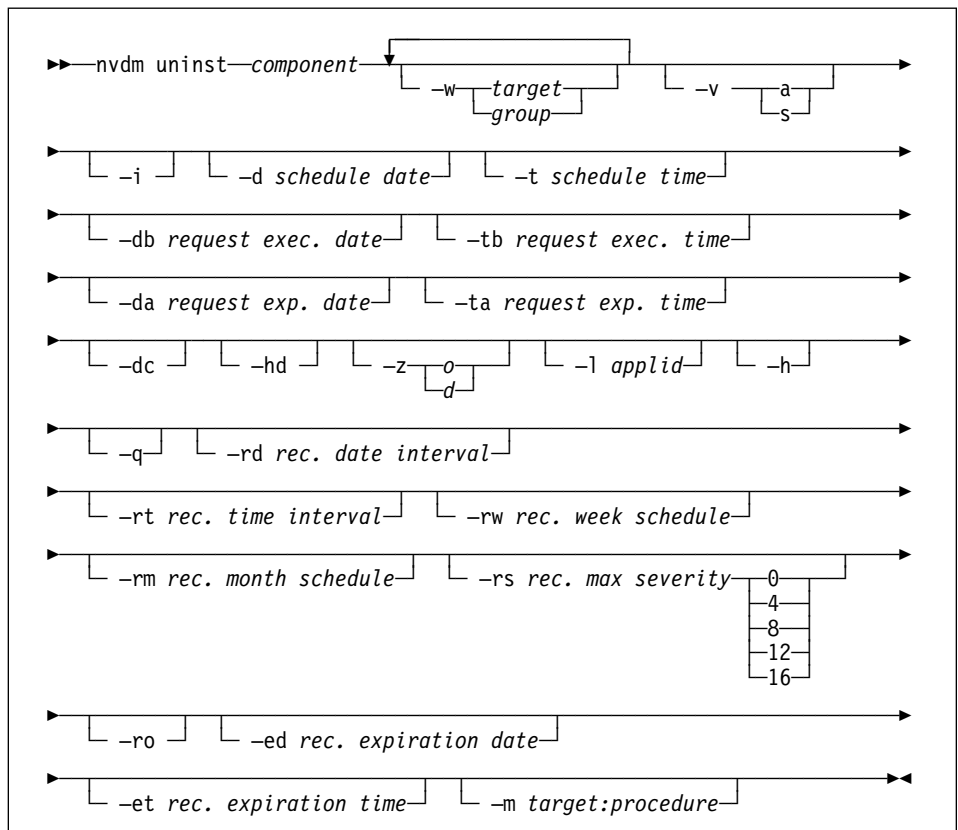
Use the **uninst** command to remove a previously installed component from a target. The component may have originally been built from the installation of multiple change files, all of which are removed.

This command, however, does not uninstall a file included in a change file if the change file was created and installed in the following way:

1. A file called `fred` exists on a target. Another file with the same name, `fred`, is contained within a change file and it is installed on that target using TME 10 Software Distribution. The existing `fred` is overwritten by this installation.
2. When the **uninst** command is issued for the change file that includes `fred`, the command is run but the `fred` file presently stored on the target is not erased.

This command can be used locally by mobile clients. It does not, however, accept the `-w`, `-h`, `-z`, `-rd`, `-rt`, `-rw`, `-rm`, `-rs`, `-ro`, `-ed`, `-et` or `-m` parameters.

Syntax



Parameters

component

Component name of the component to uninstall. You cannot use wildcard characters.

-w *target | group*

The targets or groups to which this command applies. See “Entering Target Names” on page 12.

This parameter cannot be used locally by mobile clients.

-v *ajs*

If **-va** is specified, the component is removed from the active area immediately. If **-vs** is specified it removes the component from the active area on receipt of a subsequent Activate request. If neither is specified, the default is **-va**.

-i

Force the Uninstall command to be scheduled even if the change file history on the target does not allow it. However, the uninstallation will not succeed if the change file history forbids it at the time when the uninstallation is to be executed.

-d *schedule date*

The date the request is to take place at the origin target.

Date and time parameters are described in more detail in “Entering Dates and Times” on page 13.

-t *schedule time*

The time the request is to take place at the origin target.

-db *request execution date*

The date the request is to take place at the destination target.

-tb *request execution time*

The time the request is to take place at the destination target.

-da *request expiration date*

The date after which the request cannot take place at the destination target.

-ta *request expiration time*

The time after which the request cannot take place at the destination target.

-dc

This parameter refers to operations performed at mobile clients. When it is specified, the operation is to be performed when the mobile client is disconnected from the server.

-hd

This parameter pertains to operations performed at mobile clients, and can only be specified together with the **-dc** parameter. It holds the disconnected request at the server until its execution time is reached. The request is forwarded to the client during the next connection window opened after the execution time specified, and executed there when the client is disconnected.

-z o|d

The format of the execution time. Enter one of the following values:

- o** The time is expressed as the local time at the origin target. This is the default.
- d** The time is expressed as the local time at the destination target.

This parameter cannot be used locally by mobile clients.

-l *application id*

The ID of the application used to submit the request. This ID can be used later to filter the requests submitted by a specific application.

It does not make sense to set an application ID for a request issued from the graphical interface, as this is an interactive process. Requests submitted from the graphical interface are automatically given the value of GUI as the application ID. You cannot change this value.

If you do not specify an application ID for requests issued from the command-line interface, the default value is CLI.

- h** The request is to be submitted with high priority. You cannot specify this parameter if schedule times and execution times are specified.

This parameter cannot be used locally by mobile clients.

- q** The request is to be submitted in held state. The request is scheduled only when released by the operator.

This parameter cannot be used locally by mobile clients.

-rd *rec. date interval*

A recursion date interval expressed in years, months, days. The request is executed upon every lapse of this interval. To specify an interval in terms of days, specify also the months and the years. To specify an interval in terms of months, specify also the years.

For example, 1 means the request is executed every year; 0,1 means the request is executed every month; 0,2,1 means the request is executed every day of every other month.

This parameter cannot be used locally by mobile clients.

-rt *rec. time interval*

A recursion time interval expressed in hours and minutes. The request is executed upon every lapse of this interval. The format of a time interval is the same as for a time. For example, 3:45 means a time interval of 3 hours and 45 minutes.

This parameter cannot be used locally by mobile clients.

-rw *rec. week schedule*

The days of the month when the recursive request is to take place. The days of the week are as follows:

0 = Sunday
 1 = Monday
 2 = Tuesday
 3 = Wednesday
 4 = Thursday
 5 = Friday
 6 = Saturday

For example, to set a recursion for each Monday and Friday of the week, enter:

```
-rw 1,5
```

This parameter cannot be used locally by mobile clients.

-rm *rec. month schedule*

The days of the month when the recursive request is to take place. Each day of the month is represented by a number from 1 to 31.

For example, to select a recursion for the 15th and 27th of the month, enter:

```
-rm 15,27
```

The number 31 specifies the last day of any month. The number 29 specifies the last day of February even in a non-leap year.

This parameter cannot be used locally by mobile clients.

-rs *rec. max severity* **0|4|8|12|16**

A new occurrence of the request is generated only if the severity level returned from the previous occurrence is less than or equal to the max-severity specified.

Specify one of the following:

0 Successful
4 Warning
8 Error
12 Failed
16 Unrecoverable

This parameter cannot be used locally by mobile clients.

-ro Specify this parameter to stop the recursion mechanism if the current occurrence of the request is in progress when the new occurrence should be generated. If you do not specify this parameter, one or more occurrences may be skipped until the current one is completed, but the recursion mechanism is not stopped. This parameter is not set by default.

This parameter cannot be used locally by mobile clients.

uninst

-ed *rec. expiration date*

The date after which the system stops the recursion mechanism.

This parameter cannot be used locally by mobile clients.

-et *rec. expiration time*

The time after which the system stops the recursion mechanism.

This parameter cannot be used locally by mobile clients.

-m *target:procedure*

A cataloged procedure to execute on a given target when the request is completed. See “Running a Procedure after a Request Has Completed” on page 16.

This parameter cannot be used locally by mobile clients.

Examples

```
nvdm uninst EURO.WORDPROC -w FREDSW -va -d “3/1/00” -t “2:00”
```

This command uninstalls the component EURO.WORDPROC from target FREDSW at the time and date specified, if the system is quiesced at that time.

Related Information

The inst and rem commands.

upd – Update a Catalog Entry

Use the **upd** command to update a catalog entry.

This command cannot be used at mobile clients when working with the local catalog.

Syntax

```
►►—nvdm upd—global name—└—kd access key—◄◄
```

Parameters

-kd *access key*

The data access key associated with the file. The access key name specified must already be defined.

Examples

```
nvdm upd EURO.WORDPROC.REF.2.2 -kd DAK20
```

This command updates the change file EURO.WORDPROC.REF.2.2 in the catalog by associating data access key DAK20 with it.

Related Information

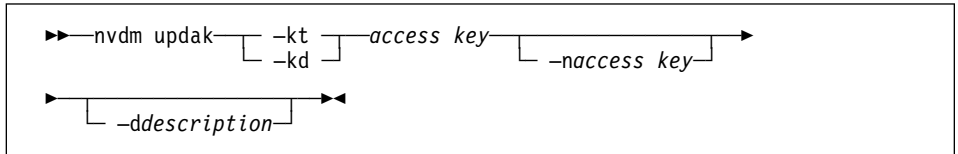
The cat command.

updak

updak – Update an Access Key

Use the **updak** command to update a data or a target access key. You must have Modify System Administration authorization to use this command.

Syntax



Parameters

-kt The key to be modified is a target access key.

-kd

The key to be modified is a data access key.

access key

The name of the access key to be modified.

-n *access key*

The new name for the access key.

-d *description*

The description associated to the access key. If a description is already associated with the key, this new one overrides it. The description can be up to 59 characters long.

Examples

```
nvdm updak -kt TAK01 -n TAK10 -d Access key for builder sites.
```

This command updates the target access key TAK01. Its name changes to TAK10 because the original definition was incorrect, and adds a description to it.

Related Information

The `lsak` command.

updbbs – Update Configuration

Use the **updbbs** command to update configurations with the base and server options installed. You must have Modify System Administration authorization to use it.

The only configuration information that is available for you to update is the TCP/IP authorization level used. This command is not available in single-node systems, because there are no TCP/IP-attached clients.

Syntax

```
nvdm updbbs -a v|n
```

Parameters

-a v|n

The LAN address can be verified for each application using RBAPI. Specify one of the following:

-av TCP/IP network addresses are verified

-an TCP/IP network addresses are not verified

Examples

```
nvdm updbbs -av
```

This command starts a TCP/IP check of network addresses in configurations with the base and server options installed.

Related Information

The lsbs command.

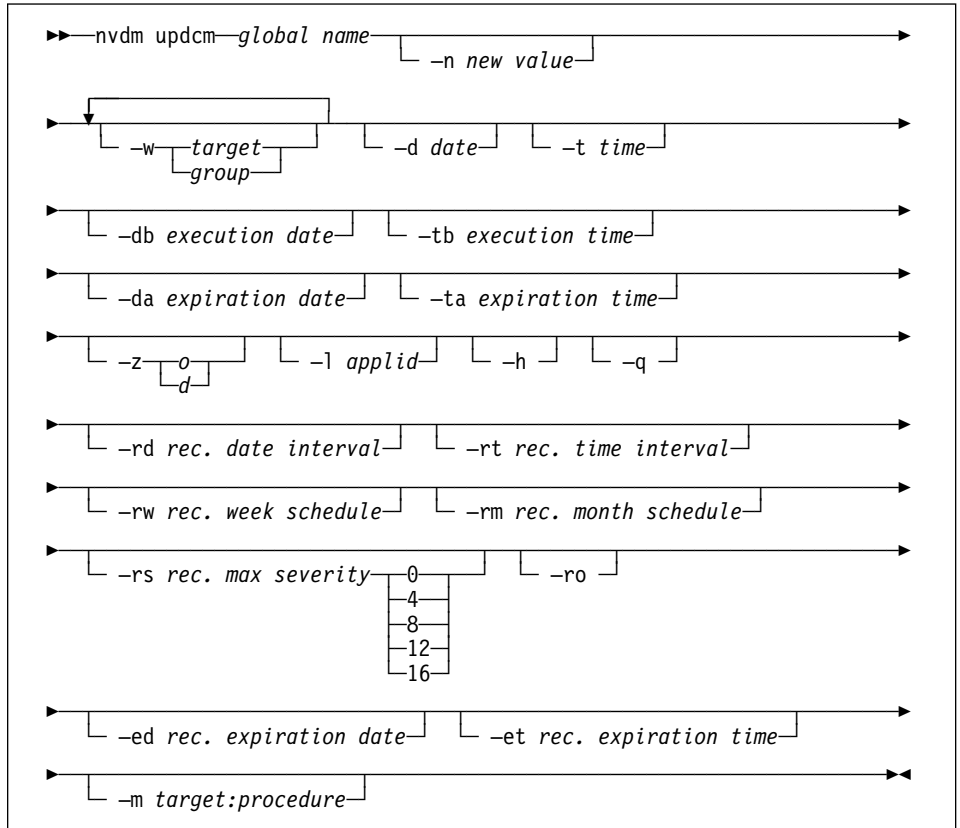
updcn – Update Change Control Status

Use the **updcn** command to add or modify the status associated with an object in the server catalog, without actually performing the action described by the status. You can issue this command for a change file, a data file, or a plan. You must have authorization for the Authorization/Unauthorization functions in your user profile to work with this command. This command is useful in situations such as:

- When targets are fully disconnected, and thus are never directly connected with a server. On fully disconnected targets, users install software from diskettes; an administrator must thus manually keep track of software history on these targets by entering history manually in the catalog. The **updcn** command can be used to modify status information.
- When change control statuses at the server do not reflect the actual situation at targets. **updcn** can be used in such a situation to set the status correctly.

The **updcn** command always refers to a single object. Therefore, when you issue this command for a change file that has corequisites, the change control status of the corequisites is not modified.

Syntax



Parameters

global name

The global name of the file whose status is to be modified. See "Entering Global Names" on page 13. You can use wildcard characters, but a unique match must be found.

This parameter is required, unless either "activated" or "rebooted" are specified as the **new value**, because they refer to targets, for which no global name is required.

-n new value

The new value for the change control status. This parameter is required. Enter one of the following values:

Acronym

activated
available
discovered
distributed
inra
inri

Stands For

activation for the target (see below)
available
discovered
distributed
installed, not removable, active
installed, not removable, inactive

ira	installed, removable, active
iri	installed, removable, inactive
irr	installed, removable, reboot required
inrr	installed, not removable, reboot required
rebooted	reboot for the target (see below)
ri	removed, inactive
ui	uninstalled, inactive

Note that for objects that are not change files (such as data files and plans), the only applicable statuses are "distributed" and "available".

-w *target | group*

The targets or groups where the change control status is to be modified. This parameter is optional. If it is not specified, the target is assumed to be the one you are issuing the command from. See "Entering Target Names" on page 12.

-d *date*

The date the request is to take place at the origin target.

Date and time parameters are described in more detail in "Entering Dates and Times" on page 13.

-t *time*

The time the request is to take place at the origin target.

-db *request execution date*

The date the request is to take place at the destination target.

-tb *request execution time*

The time the request is to take place at the destination target.

-da *request expiration date*

The date after which the request cannot take place at the destination target.

-ta *request expiration time*

The time after which the request cannot take place at the destination target.

-z *o|d*

The format of the execution time. Enter one of the following values:

- o** The time is expressed as the local time at the origin target. This is the default.
- d** The time is expressed as the local time at the destination target.

-h The request is to be submitted with high priority. You cannot specify this parameter if schedule times and execution times are specified.

-q The request is to be submitted in held state. The request is scheduled only when released by the operator.

This parameter cannot be used locally by mobile clients.

-rd *rec. date interval*

A recursion date interval expressed in years, months, days. The request is executed upon every lapse of this interval. To specify an interval in terms of days,

specify also the months and the years. To specify an interval in terms of months, specify also the years.

For example, 1 means the request is executed every year; 0,1 means the request is executed every month; 0,2,1 means the request is executed every day of every other month.

This parameter cannot be used locally by mobile clients.

-rt *rec. time interval*

A recursion time interval expressed in hours and minutes. The request is executed upon every lapse of this interval. The format of a time interval is the same as for a time. For example, 3:45 means a time interval of 3 hours and 45 minutes.

This parameter cannot be used locally by mobile clients.

-rw *rec. week schedule*

The days of the week when the recursive request is to take place. The days of the week are as follows:

0 = Sunday	4 = Thursday
1 = Monday	5 = Friday
2 = Tuesday	6 = Saturday
3 = Wednesday	

For example, to set a recursion for each Monday and Friday of the week, enter:

```
-rw 1,5
```

This parameter cannot be used locally by mobile clients.

-rm *rec. month schedule*

The days of the month when the recursive request is to take place. Each day of the month is represented by a number from 1 to 31.

For example, to select a recursion for the 15th and 27th of the month, enter:

```
-rm 15,27
```

The number 31 specifies the last day of any month. The number 29 specifies the last day of February even in a non-leap year.

This parameter cannot be used locally by mobile clients.

-rs *rec. max severity 0|4|8|12|16*

A new occurrence of the request is generated only if the severity level returned from the previous occurrence is less than or equal to the max-severity specified.

Specify one of the following:

0	Successful
4	Warning
8	Error
12	Failed
16	Unrecoverable

This parameter cannot be used locally by mobile clients.

updcn

- ro** Specify this parameter to stop the recursion mechanism if the current occurrence of the request is in progress when the new occurrence should be generated. If you do not specify this parameter, one or more occurrences may be skipped until the current one is completed, but the recursion mechanism is not stopped. This parameter is not set by default.

This parameter cannot be used locally by mobile clients.

- ed** *rec. expiration date*

The date after which the system stops the recursion mechanism.

This parameter cannot be used locally by mobile clients.

- et** *rec. expiration time*

The time after which the system stops the recursion mechanism.

- l** *application id*

The ID of the application used to submit the request. This ID can be used later to filter the requests submitted by a specific application.

It does not make sense to set an application ID for a request issued from the graphical interface, as this is an interactive process. Requests submitted from the graphical interface are automatically given the value of GUI as the application ID. You cannot change this value.

If you do not specify an application ID for requests issued from the command-line interface, the default value is CLI.

- m** *target:procedure*

A cataloged procedure to execute on a given target when the request is completed. See "Running a Procedure after a Request Has Completed" on page 16.

Examples

```
nvdm updcn EURO.WORDPROC.REF.1.1.0.0 -n ira -w fredsws
```

This command updates the status of the change file EURO.WORDPROC.REF.1.1.0. at the target fredsws to "installed, removable, active".

Related Information

The delcm, lscm, and vercm commands.

updpln – Update a Plan

Use the **updpln** command to update a plan that is cataloged.

Syntax

```
►►—nvdm updpln—profile name—◄◄
```

Parameters

profile name

The name of the plan profile.

Examples

```
nvdm updpln EUROPROFILE
```

This command updates the plan in the catalog that corresponds to the profile EUROPROFILE.

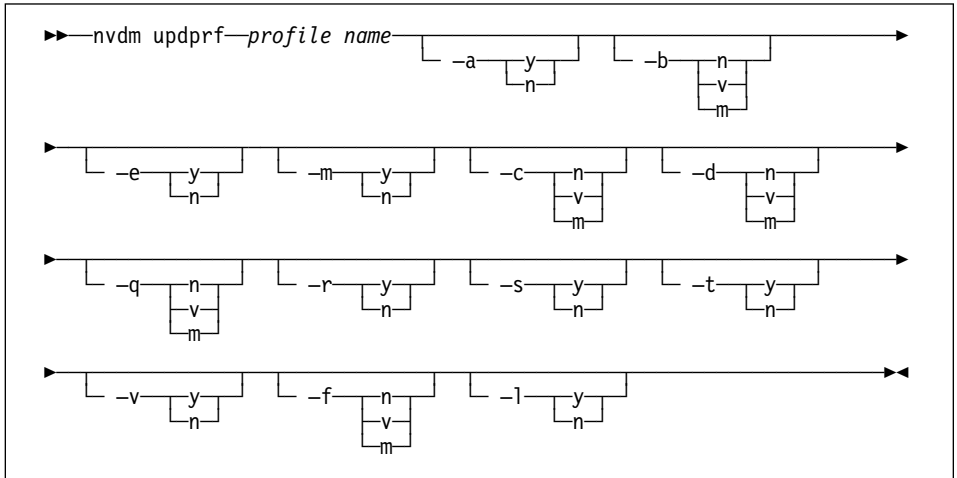
Related Information

The addpln command.

updprf – Update a User Authorization Profile

Use the **updprf** command to update a user authorization profile. To be updated, a profile must include at least one meaningful parameter.

Syntax



Parameters

If you do not include one of the following parameters, the current settings for the parameter remain unchanged.

profile name

The name of the profile to be changed.

-a y|n

Change control authorization operations. Allows access to the following functions:

- Authorize
- Unauthorize
- Delete History

It can be set to:

- y** Allow access to change control authorization functions.
- n** Deny access to change control authorization functions.

-b n|v|m

Build authorization. It can be set to one of the following:

- n** Deny access to build information.
- v** Allow information about the building of change files to be viewed.
- m** Allow all operations including the building of change files.

-e y|n

Change control execute authorization. It can be set to one of the following:

- y** Allow access to change control execute function.
- n** Deny access.

-m y|n

Change control operations. It allows access to the following functions:

- Install
- Remove
- Accept
- Uninstall
- Activate

It can be set to:

- y** Allow access to the change control functions listed above.
- n** Deny access to the change control functions listed above.

-c n|v|m

Configuration authorization. It can be set to one of the following:

- n** Deny access to configuration information.
- v** Allow configuration information to be viewed.
- m** Allow all operations including the modification of configuration information.

-d n|v|m

System administration authorization. It can be set to one of the following:

- n** Deny access to system administration information.
- v** Allow system administration information to be viewed.
- m** Allow all operations including modification of system administration information.

-q n|v|m

Queue operations. It can be set to one of the following:

- n** Deny access to queue information.
- v** Allow queue information to be viewed.
- m** Allow all operations including control of queues.

-r y|n

Retrieve, delete, and replace authorization. It can be set to one of the following:

- y** Allow access to the retrieve, delete, and replace functions.
- n** Deny access to the retrieve, delete, and replace functions.

-s y|n

Send authorization. It can be set to one of the following:

- y** Allow access to the send function.
- n** Deny access to the send function.

updprf

-t y|n

Target management authorization. It can be set to one of the following:

- y** Can perform change control at all targets.
- n** Can manage own target only.

-v y|n

Change management activation operation. It can be set to one of the following:

- y** Allow access to the activation function.
- n** Deny access to the activation function.

-f n|v|m

Request management. It can be set to one of the following:

- n** No request commands can be issued.
- v** Request information can be displayed.
- m** All request operations can be performed.

-l y|n

Manage all requests flag. It can be set to one of the following:

- y** Request commands can be performed on all requests.
- n** Request commands can be performed only on the user's own requests.

Examples

```
nvdm updprf EMPLOYEE -my -bn -qv
```

This command specifies that users associated with the profile EMPLOYEE have authorization to the following functions:

- Change control
- No build functions
- View queues

All other authorization is unchanged.

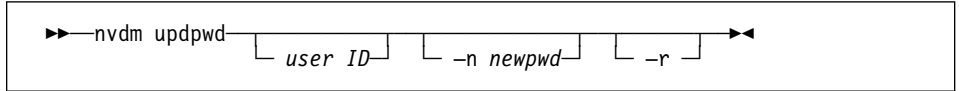
Related Information

The lsusr, lstg, addpm, stattg, and addtg commands.

updpwd – Update a User Password

Use the **updpwd** command to update a user's password.

Syntax



Parameters

-user ID

The user's identifier. If none is specified, the default is the user ID you are using.

-n *newpwd*

The user's new password. It must be between six and eight characters, entered in the form:

pwd/pwd

-r Resets the password to none. If you specify this parameter, the user can log on entering the user ID only.

Validation Check

Only the user root can change the password set for this user.

If you enter this command without parameters, you are prompted to enter them.

Examples

```
nvdm updpwd JANEDOE -n hello/hello
```

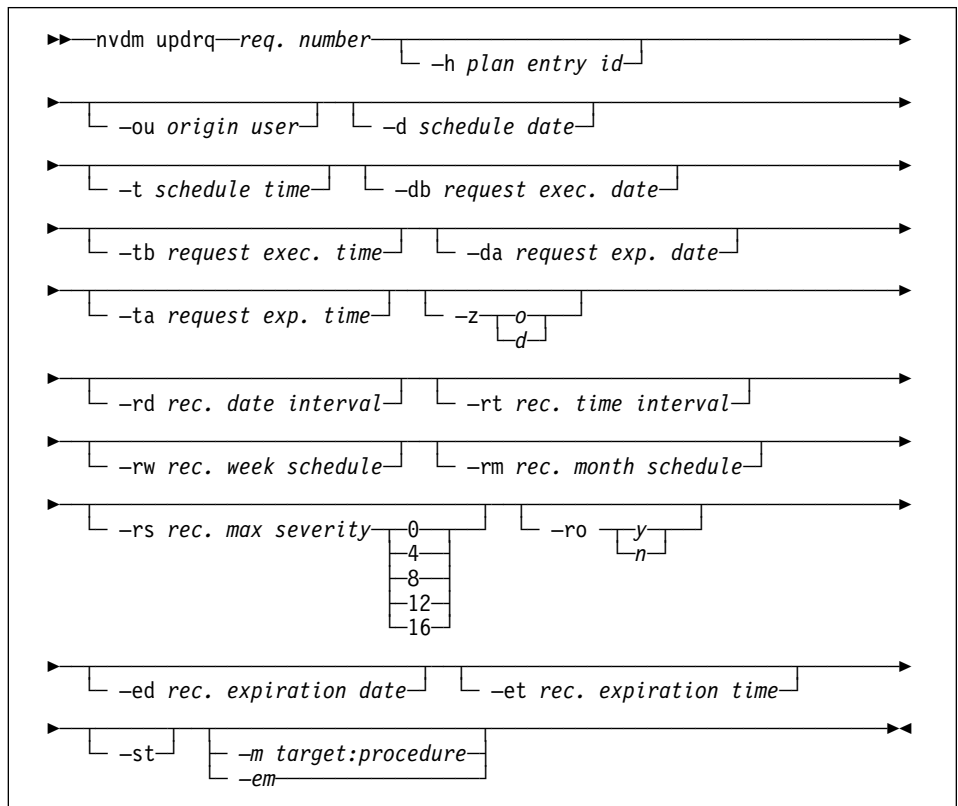
This command updates the password for user JANEDOE.

updrq – Update a Request

Use the **updrq** command to update the parameters set for a request. If the request refers to a plan, new date and time must not conflict with those set for any entries in the plan.

You can change all parameters, except **-ro** and **-rs**, if the request is in the waiting or held state. If the request refers to a plan, parameters can be changed for a plan entry that is in the held or waiting state even though the plan execution is in progress. You can update the **-ro** and **-rs** parameters even if the request is in progress; the new values are effective from the following occurrence.

Syntax



Parameters

request number

The sequential number of the request. This parameter is required.

-h *plan entry id*

The identification of the plan entry. If the request does not refer to a plan, an error message is displayed.

-ou *origin user*

The user who submitted the request. Only the administrator can update the scheduling of requests that were submitted by other users. This parameter is optional. The default is the name of the user submitting this command.

-d *schedule date*

The date on which the request is to be executed at the origin target.

Date and time parameters are described in more detail in “Entering Dates and Times” on page 13.

-t *schedule time*

The time at which the request is to be executed at the origin target.

-db *request execution date*

The date on which the request is to be executed at the destination target.

-tb *request execution time*

The time at which the request is to be executed at the destination target.

-da *request expiration date*

The date after which the plan or request cannot be executed at the destination target.

-ta *request expiration time*

The time after which the plan or request cannot be executed at the destination target.

-z o|d

The format of the execution time. Enter one of the following values:

- o The time is expressed as the local time at the origin target. This is the default.
- d The time is expressed as the local time at the destination target.

-rd *rec. date interval*

A recursion date interval expressed in years, months, days. The request is executed upon every lapse of this interval. To specify an interval in terms of days, specify also the months and the years. To specify an interval in terms of months, specify also the years.

For example, 1 means the request is executed every year; 0,1 means the request is executed every month; 0,2,1 means the request is executed every day of every other month.

-rt *rec. time interval*

A recursion time interval expressed in hours and minutes. The request is executed upon every lapse of this interval. The format of a time interval is the same as for a time. For example, 3:45 means a time interval of 3 hours and 45 minutes.

-rw *rec. week schedule*

The days of the week when the recursive request is to take place. The days of the week are as follows:

- 0 = Sunday
- 1 = Monday
- 2 = Tuesday
- 3 = Wednesday
- 4 = Thursday
- 5 = Friday
- 6 = Saturday

For example, to set a recursion for each Monday and Friday of the week, enter:

```
-rw 1,5
```

-rm *rec. month schedule*

The days of the month when the recursive request is to take place. Each day of the month is represented by a number from 1 to 31.

For example, to select a recursion for the 15th and 27th of the month, enter:

```
-rm 15,27
```

The number 31 specifies the last day of any month. The number 29 specifies the last day of February even in a non-leap year.

-rs *rec. max severity* **0|4|8|12|16**

A new occurrence of the request is generated only if the severity level returned from the previous occurrence is less than or equal to the max-severity specified.

Specify one of the following:

- 0** Successful
- 4** Warning
- 8** Error
- 12** Failed
- 16** Unrecoverable

-ro *y|n*

Stop the recursion mechanism on overlap. Enter one of the following values:

- y** The recursion mechanism is stopped if the current occurrence of the request is in progress when the new occurrence should be generated.
- n** One or more recursion occurrences are skipped if the current one has not completed, but the recursion mechanism is not stopped.

-ed *rec. expiration date*

The expiration date of the recursion mechanism.

-et *rec. expiration time*

The expiration time of the recursion mechanism.

-st Stop the recursion mechanism.

-m *target:procedure* | **-em**

Enter one of the following parameters:

-m *target:procedure*

A cataloged procedure to execute on a given target when the request has completed. See “Running a Procedure after a Request Has Completed” on page 16.

-em

Cancel the procedure execution.

Examples

```
nvdm updrq 13 -h 00022 -ed 12/31/00
```

This command updates entry 00022 of request 13. It changes the expiration date for the entry to December 31, 2000.

Related Information

The delrq, eraserq, relrq, and rstrq commands.

updtg – Update Targets

Use the **updtg** command to update the definition of an existing target. Any options that are not explicitly updated are left with their current definitions, which were set or defaulted on the original **addtg** command.

Some options are invalidated by changing the settings of other options. For example, if you change the mode of a target from push to pull, the change control time range is no longer meaningful. The system automatically handles this by disregarding the previously set options where they conflict with new settings of options.

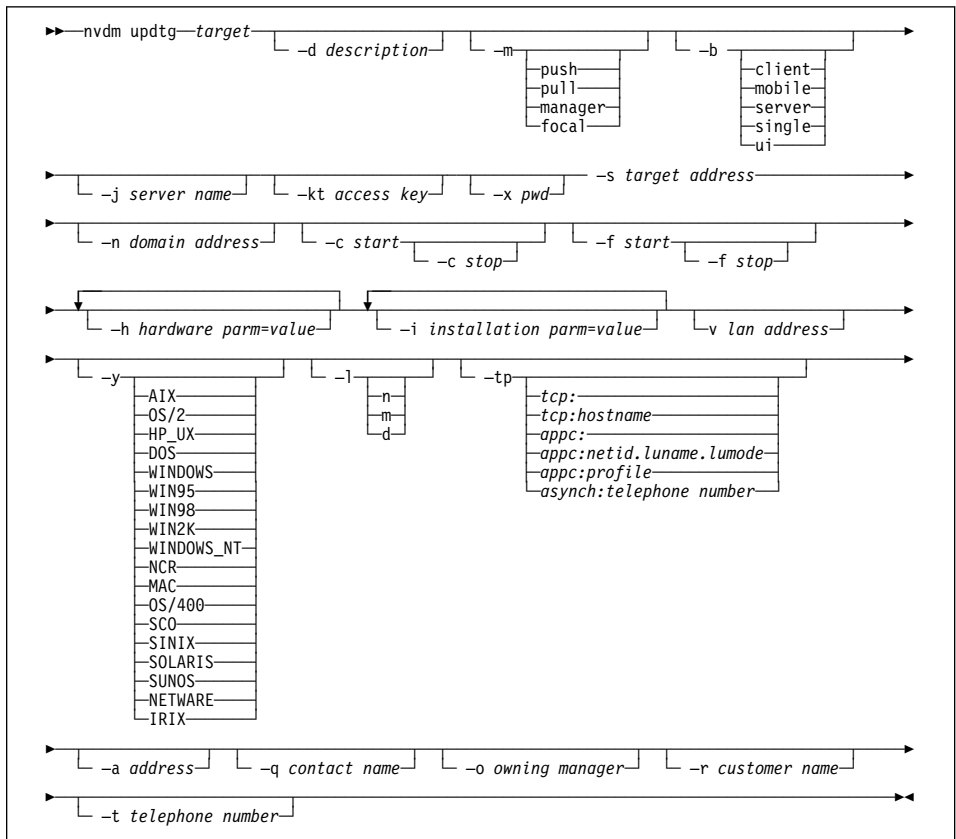
If there are pending requests for the target, you cannot change the name of the target, the server name (**-j** parameter), target address (**-s** parameter), and domain address (**-n** parameter).

You must have Modify Configuration authorization to use this command.

If you do not have Modify Configuration authorization, you can update only the definition of your own login target. You can change the following:

- Description
- Customer name
- Contact name
- Phone number
- Owning manager
- Address
- Target password

Syntax



Parameters

target

Target to be updated. The name must be specified precisely.

If you specify the name of a target which is not already defined, the command is rejected. Use the **addtg** command to define a new target.

-d *description*

A description of the target being updated. If this parameter is omitted, no description is saved.

-m *push|pull|manager|focal*

The mode in which this target can be used. The valid options are:

push	Push mode target
pull	Pull mode target
manager	Manager node
focal	Remote focal point

You can perform only limited updating of the mode of a target. Two subsets are defined. You can only change the mode of a target from one mode in a set to another mode in the same set. The sets are:

- Push mode, pull mode
- Manager mode, focal point mode

For example, if a target is defined with pull mode, you can use the **updtg** command to change its mode to push or you can leave it as pull. To change the mode to some other value, ensure that the **-b** parameter has a value that is not in conflict with the **-m** parameter.

If you change the mode of a target, it is removed from any groups to which it belonged because a group can only contain targets that have the same mode.

-b client|server|single|ui |mobile

The type of target. The valid options are:

client	The target is a client. If you specify a remote client, you must also specify the name of the server that the client is connected to using the -j option.
server	The target is a server.
single	The target is a single node.
ui	The target is user interface only.
mobile	The client is has the mobile function installed.

-j server name

The server the client is connected to. Specify this parameter only if the target type is client. The default is the server the client is physically connected to.

-kt access key

The target access key. This parameter is optional. The default is none, meaning that no access key is assigned to the target. If you specify this parameter, the access key must exist, and must be assigned to users of this target.

-x pwd

The password to be used to access the target. The password is necessary only if the TARGET PASSWORD AUTHENTICATION keyword in the server base configuration file is set to YES. If it is, then whenever a target attempts communication with its server, the target sends it its password and the server checks to make sure that it corresponds to the password it has stored in the target configuration.

The password must be between 6 and 8 characters long and entered in the form `pwd/pwd`.

This parameter is optional. If you do not specify a value, the name of the target is used.

-s target address

The address for this target. For systems with remote communication this is the REN part of the SNA/DS address. In a local network, the target address assigned to a server must also be used as the routing group name (RGN) for the server itself and all clients in its domain.

This parameter is valid for all targets.

-n domain address

This parameter is valid only for remote targets. It can be up to 8 characters long; the only valid characters are letters and digits. The value you define with this parameter depends on the type of target you are updating:

TME 10 Software Distribution server or single node

Specify the same value specified for this target's address.

TME 10 Software Distribution client

Specify the target address of the server the client is connected to.

Non TME 10 Software Distribution target

If the target you are updating is *not* a TME 10 Software Distribution target, specify the domain address of the remote network that this target belongs to.

In an SNA network, this value corresponds to the routing group name (RGN) part of the SNA/DS address of the target. Ask the administrator at the host processor what the value is.

In a TCP/IP network, this value must simply correspond to the ID of the remote network this target belongs to.

-c start [-c stop]

Period of time allocated for change control requests. If the parameter is present twice, the first occurrence is the start time and the second is the stop time. If only one time is specified, it is assumed to be the start time; the stop time is defaulted.

The start time defaults to 00:00, and the stop time to 23:59 (that is, change control operations can be performed at any time).

This parameter can be specified only for local targets. It cannot be used for UI only, remote or focal point targets.

-f start [-f stop]

Period of time allocated for distribution requests. If the parameter is present twice, the first occurrence is the start time and the second is the stop time. If only one time is specified, it is assumed to be the start time; the stop time is defaulted.

The start time defaults to 00:00, and the stop time to 23:59 (that is, distribution operations can be performed at any time).

This parameter can be specified only for local targets. It cannot be used for UI only, remote or focal point targets.

-h *hardware parm=value*

The definition of a hardware parameter that is used for checking hardware prerequisites when installing a change file (for example `-h mem=40`). To redefine an existing parameter, you must first delete it using the **delpm** command.

You can specify this parameter up to 10 times for each command. You can specify up to 128 hardware parameters for each target. This parameter can be specified only for local targets. It cannot be used for UI only, remote or focal point targets.

-i *installation parm=value*

An installation parameter to be used by the target during change file installation.

You can specify this parameter up to 10 times for each command. It can be specified only for local targets. It cannot be used for UI only, remote or focal point targets.

-v *lan address*

MAC network burned-in address for this target. The default is no address.

This parameter cannot be used for remote or focal point targets.

-y *AIX|OS/2|HP_UX|DOS|WINDOWS|WIN95|WIN98|WIN2K|WINDOWS_NT|SCO|NCR|MAC|SINIX|SOLARIS|SUNOS|NETWARE|IRIX|OS/400*

This updates the operating system type for targets. Note that WINDOWS indicates Windows 3.11, while WIN95, WIN98 and WIN2K are Windows 95, Windows 98 and Windows 2000, respectively.

-l *n|m|d*

Specifies the level of message logging to be performed. They are:

n	Normal (the default)
m	Minimal
d	Diagnostic

This parameter cannot be specified for UI only or remote targets.

-tp *protocol type*

This parameter is applicable only when STS communication is being used. It specifies the type of communication protocol used by a client to connect to the server or by a server to connect to another server.

Valid options are:

tcp: The transmission protocol is TCP/IP.

tcp:hostname

The transmission protocol is TCP/IP, with the host name you specify.

appc: The transmission protocol is APPC.

appc:netid.luname.lumode

The transmission protocol is APPC, with the specified network ID, LU name, and LU mode related to the partner.

If you specify the parameter using this format, be sure to include all of these values.

appc:profile

The transmission protocol is APPC, with the name of the SNA Server/6000 Version 2.1 side destination profile you specify.

asynch:telephone number

The transmission protocol is asynchronous, which is used to link to a mobile client. Specify the telephone number used to connect to the client.

The default values are as follows:

- If you omit this parameter, **-tp tcp:** is the default.
- If **-tp tcp:** is specified, the target name is used.
- If **-tp appc:** is specified, the **-tp appc:netid.luname.lumode** option is used, where:

The network ID defaults to the target's routing group name (RGN)

The LU name defaults to the target's routing element name (REN)

The LU mode is blank

This parameter can be used for servers and local clients that are connected using STS communication.

-a address

Contact address for the specified target. The default is blank. Use an asterisk (*) in the address to begin a new line.

-q contact name

A contact name at the target. The default is blank.

-o owning manager

Owning manager of the target. The default is blank.

-r customer name

The name of the customer at the target. The default is blank.

-t telephone number

A telephone number at the target. The default is blank.

Examples

```
nvdn updtg FREDWS -m pull -d "Fred's Workstation (pull mode)"
```

This command updates the local target FREDWS. It becomes a pull mode target. This change is reflected in a new description for the target.

```
nvdn updtg HOST -m remote -d "The host" -r "Head Office"
-q "Mr P.C. Dixon" -t "071 230 1212"
```

updtg

This command updates the target definition for the host processor at head office. This is a remote target known as HOST.

Focal point checking is no longer required, so the mode is changed to **remote**. At the same time, the contact name and telephone number are changed.

Related Information

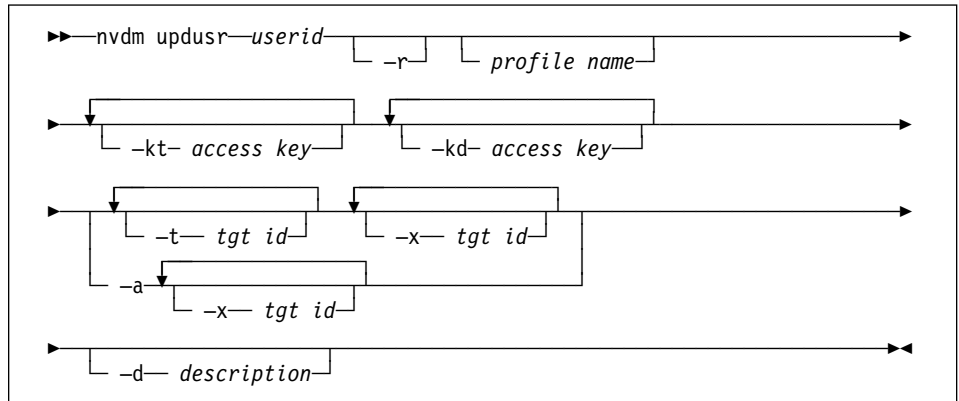
The addpm, addtg deltg, delpm, lstg, rentg, and stattg commands.

updusr – Update a User Definition

Use the **updusr** command to update the definition of a user.

You must have full System Administration authorization to use this command.

Syntax



Parameters

userid

The identifier of the user whose definition you want to update. It can have up to eight alphanumeric characters, excluding all special characters.

- r Specify this parameter to reset the number of login attempts to the value defined in the MAX_LOGIN_ATTEMPTS keyword of the configuration file.

profile name

The name of the new authorization profile you want the user to be associated with.

-kt *access key*

The target access key (TAK) associated with the user. If specified, the TAK name must exist in the access key table.

If you specify **all**, all access keys are associated with this user. If you specify **none**, all access keys are removed.

-kd *access key*

The data access key (DAK) associated with the user. If specified, the DAK name must exist in the access key table.

If you specify **all**, all access keys are associated with this user. If you specify **none**, all access keys are removed.

-t *tgt id*

The list of targets where the user can log in.

-a If this parameter is specified, the user can log in every target.

-x *tgt id*

The list of targets where the user cannot log in.

-d *description*

A description of the user being defined. If the description contains blanks or any special characters that might be interpreted by the command processor, it must be enclosed by delimiters (usually double quotation marks).

You can enter up to 59 characters.

Note that for target authorizations the following rules apply:

Table 10. Updating User Authorizations to Targets						
Flags Previous Settings		Command Parameters				
A	X	-a	-t	-x	-a & -x	-t & -x
0	0	A=1, tgt_id list is erased	tgt_ids are added to tgt_id list	matching tgt_ids in list are erased	A=1, tgt_id list is replaced	tgt_ids with -t are added, with -x are erased
1	0	nothing happens	A=0, tgt_id list is added	X=1,tgt_id list is added	X=1, tgt_id list is added	A=0, tgt_ids with -t are added
1	1	X=0, tgt_id list is erased	matching tgt_ids in list are erased	tgt_ids are added to tgt_id list	tgt_ids are added to tgt_id list	tgt_ids with -x are added, with -t are erased

The effects on both the flags (A and X) and the target list are shown. Two simple rules to keep in mind are:

- If you specify -t, make that target be one where the user can log in.
- If you specify -x, exclude the target from those where the user can log in.

Examples

```
nvdms updusr JOHND0E -kt TAK04 -a -x FREDWS
```

This command updates the definition of user JOHND0E. The target access key is changed to TAK04, and the user is given access to all targets except for FREDWS.

Related Information

The addusr, delusr, and lsusr commands.

vercm – Verify Change Control Status

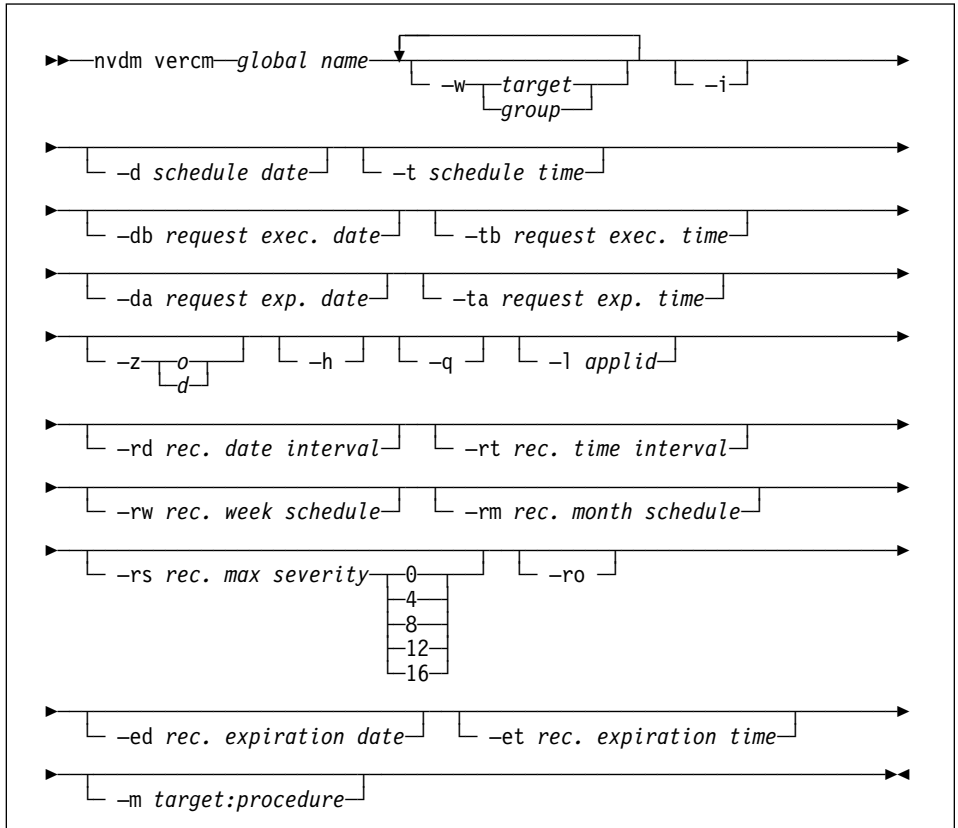
Use the **vercm** command to verify:

- The consistency between the change control status of one or more installp change files in the TME 10 Software Distribution database, and the native database
- If all the files installed with a generic change file are present at a target

When you verify an installp change file, TME 10 Software Distribution changes the change control status to *in error*. If the change control status is different from the status found in the native database, TME 10 Software Distribution adds *in error* to the change control status. For instance, if the change control status is *installed*, *not removable*, *active* and you remove some of the filesets installed with the change file, you create an inconsistency between the entries in the TME 10 Software Distribution and native databases.

When you verify a generic change file, if TME 10 Software Distribution detects that files in the change file (specified as COPY FILE or COPY REMOTE FILE) are missing at a target, it adds *in error* to the change control status. A report file called FNDSWCMS is created and stored at the target. You can retrieve the report using the **rtrv** command.

Syntax



Parameters

global name

The global name to be used in the search. You can use wildcard characters.

-w *target | group*

The targets or groups to search. See “Entering Target Names” on page 12.

- i** Force the verify command to be scheduled even if the change file history on the target does not allow it. However, the verification will not succeed if the change file history does not allow it at the time when the verification is to be done.

-d *schedule date*

The date the request is to be executed at the origin target.

Date and time parameters are described in detail in “Entering Dates and Times” on page 13.

-t *schedule time*

The time the request is to be executed at the origin target.

-db *request execution date*

The date the request is to be executed at the destination target.

-tb *request execution time*

The time the request is to be executed at the destination target.

-da *request expiration date*

The date after which the request cannot be executed at the destination target.

-ta *request expiration time*

The time after which the request cannot be executed at the destination target.

-z old

The format of the execution time. Enter one of the following values:

- o** The time is expressed as the local time at the origin target. This is the default.
- d** The time is expressed as the local time at the destination target.

-h The request is to be submitted with high priority. You cannot specify this parameter if schedule times and execution times are specified.**-q** The request is to be submitted in held state. The request is scheduled only when released by the operator.

This parameter cannot be used locally by mobile clients.

-l *application id*

The ID of the application used to submit the request. This ID can be used later to filter the requests submitted by a specific application.

It does not make sense to set an application ID for a request issued from the graphical interface, because this is an interactive process. Requests submitted from the graphical interface are automatically given the value of GUI as the application ID. You cannot change this value.

If you do not specify an application ID for requests issued from the command-line interface, the default value is CLI.

-rd *rec. date interval*

A recursion date interval expressed in years, months, days. The request is executed upon every lapse of this interval. To specify an interval in terms of days, specify also the months and the years. To specify an interval in terms of months, specify also the years.

For example, 1 means the request is executed every year; 0,1 means the request is executed every month; 0,2,1 means the request is executed every day of every other month.

-rt *rec. time interval*

A recursion time interval expressed in hours and minutes. The request is executed upon every lapse of this interval. The format of a time interval is the same as for a time. For example, 3:45 means a time interval of 3 hours and 45 minutes.

-rw *rec. week schedule*

The days of the week when the recursive request is to take place. The days of the week are as follows:

- 0 = Sunday
- 1 = Monday
- 2 = Tuesday
- 3 = Wednesday
- 4 = Thursday
- 5 = Friday
- 6 = Saturday

For example, to set a recursion for each Monday and Friday of the week, enter:

```
-rw 1,5
```

-rm *rec. month schedule*

The days of the month when the recursive request is to take place. Each day of the month is represented by a number from 1 to 31.

For example, to select a recursion for the 15th and 27th of the month, enter:

```
-rm 15,27
```

The number 31 specifies the last day of any month. The number 29 specifies the last day of February even in a non-leap year.

-rs *rec. max severity* **0|4|8|12|16**

A new occurrence of the request is generated only if the severity level returned from the previous occurrence is less than or equal to the max-severity specified.

Specify one of the following:

- 0** Successful
- 4** Warning
- 8** Error
- 12** Failed
- 16** Unrecoverable

-ro Specify this parameter to stop the recursion mechanism if the current occurrence of the request is in progress when the new occurrence should be generated. If you do not specify this parameter, one or more occurrences may be skipped until the current one is completed, but the recursion mechanism is not stopped. This parameter is not set by default.

-ed *rec. expiration date*

The expiration date of the recursion mechanism.

-et *rec. expiration time*

The expiration time of the recursion mechanism.

-m *target:procedure*

A cataloged procedure to execute on a given target when the request is completed. See "Running a Procedure after a Request Has Completed" on page 16.

Examples

```
nvdms vercm EURO.WORDPROC.1.1 -w group1 -w group2 -i -t 6:00 -rw 1
```

This command performs the change management status verification process on change file EURO.WORDPROC.1.1 at the targets in group1 and group2 every Monday at 6 a.m. The procedure is performed even if the change file history at the target does not allow it.

Chapter 3. Creating Change File Profiles

To build a change file using the command line interface **bld** command, you must first create a change file *profile*. A profile is an ASCII text file that specifies the contents of a change file. You can use any standard text editor to create it. The **bld** command uses it as input to create an actual change file.

Profiles can be prepared for *generic* change files, which can be installed on workstations running any type of operating system, or for *installp* change files, which are specific to the AIX operating system. Any specific *installp* keywords are marked by an asterisk (*).

Static and Dynamic Change File Profiles

The content of your change file (and thus your change file profile) can be *static*, meaning that the installation it triggers is always the same regardless of the workstation it is installed on, or *dynamic*, meaning that the installation is different depending on the configuration of the workstation it is being performed on. For example, the same change file could contain two different `config.sys` files: one which is installed on OS/2 workstations, and another which is installed on Windows workstations.

Dynamic change files are defined by expressing conditions that must be satisfied in order for a corresponding operation to take place. See the description of the `CONDITION` keyword on page 251, and “Dynamic Section Keywords” on page 264 for more information.

The Structure of a Change File Profile

A change file profile is divided logically into sections each of which contains a series of keywords, as shown in Figure 1 on page 248.

The Structure of a Change File Profile

ENVELOPE SECTION (of which there may be only one)

GLOBAL NAME: <name of change file>
DESCRIPTION: <description of change file>
CHANGE FILE TYPE: <type of change file>
COMPRESSION TYPE: <compression method>
SECURE PACKAGE <secure package>
REBOOT REQUIRED <reboot required>
CONDITION <condition to apply>

SOFTWARE PREREQUISITE: <required software>
HARDWARE PREREQUISITE: <required hardware>

PRODUCT SECTION (of which there may be many)

PRODUCT:
TAG: <label for product>
REVISION: <revision level>
ARCHITECTURE: <platform architecture>
VENDOR TAG: <vendor name>
TITLE: <product title>
DESCRIPTION: <description>
COST: <cost of product>

FILESET: <one for each file in product>
TAG: <label for file>
REVISION: <revision level>
TITLE: <title of file>

REMOTE DIRECTORY SECTION (of which there may only one, with up to 16 directories)

REMOTE DIRECTORY:
SERVER NAME: <server name>
EXPORTED DIRECTORY: <full directory path>
MOUNTED FILE SYSTEM: <file system>

Figure 1 (Part 1 of 2). The structure of a Change File profile

DYNAMIC SECTION (of which there may be many)	
DYNAMIC SECTION:	<name of section>
CONDITION:	<condition to apply>
SOFTWARE PREREQUISITES:	<software prerequisite>
REMOTE DIRECTORY	<name of remote source directory>
PREREQ COMMAND	<prereq command>
POSTREQ COMMAND	<postreq command>
PRE-INSTALL	<pre-install script or procedure>
POST-INSTALL	<post-install script or procedure>
PRE-ACCEPT	<pre-accept script or procedure>
POST-ACCEPT	<post-accept script or procedure>
PRE-REMOVE	<pre-remove script or procedure>
POST-REMOVE	<post-remove script or procedure>
PRE-UNINSTALL	<pre-uninstall script or procedure>
POST-UNINSTALL	<post-uninstall script or procedure>
OBJECT SECTION (of which there may be many)	
OBJECT:	
SOURCE NAME:	<name of file at source>
TARGET NAME:	<name of file at target>
TYPE:	<type of object>
ACTION:	<type of action>

Figure 1 (Part 2 of 2). The structure of a Change File profile

Where:

ENVELOPE SECTION

The overall description of the change file.

PRODUCT SECTION

Descriptions of the product and the files that compose it.

REMOTE DIRECTORY SECTION

Specifications of remote locations where files are stored.

DYNAMIC SECTION

The description of information used only under certain conditions in the change file.

OBJECT SECTION

The description of the files and other objects in the change file.

The keywords that you can include in each section of a change file profile are described below.

Entering Lines in a Profile

Each line of the profile must be less than 290 characters long, and meet one of these criteria:

- The line is blank (ignored).
- The line starts with # (treated as a comment).
- The line is in the format *KEYWORD: value*.

Only lines containing a keyword and a value are used when the change file is built from the profile. Valid keywords and values are described in “Change–File Profile Keywords.”

Rules for Entering Keywords

When you enter keywords, the following rules apply:

- Type the *KEYWORD* in uppercase followed by a colon.
- You can place any number of spaces either at the beginning of the line or between the keyword and the value.
- You do not need to include optional keywords in the profile.
- Enter all keywords included in the profile followed by a value.

Change–File Profile Keywords

This section describes the keywords used in change file profiles. An **(m)** near a keyword indicates that it is mandatory. No (m) means that keywords and their values can be omitted.

Note that when you create a change file profile for AIX installp, each keyword you use must refer to an AIX installation Image and (optionally) to a script. Each image can contain a base-level product with options, a single update, multiple updates (also referred to as packaging PTFs), or a Preventive Maintenance Package (PMP). Any mixture of images can be included in the change file.

Envelope Section Keywords

GLOBAL NAME (m)

Global name of the change file to be built.

DESCRIPTION

Description of the change file.

LOCAL NAME

Local file name for the change file when it is built. Its syntax format corresponds to *File System ID = 0*.

CHANGE FILE TYPE (m)

The type of change file. Enter this keyword in uppercase before the **DISK SPACE CHECK** keyword.

You can specify any of the following change file types, depending on the platform you are working on:

- **GEN**, used for generic change files
- **AIXINSTP**, used for AIX change files
- **OS2CID**, used for OS/2 CID change files
- **DOSCID**, used for *existing* DOS CID change files
- **WINCID**, used for *existing* WINDOWS CID change files

PACK FILES

Whether the files within the change file should be compressed. Permitted values are **YES** and **NO**. The default is **NO**.

COMPRESSION TYPE

The compression technique to use if the change file is compressed when being distributed to remote targets. Permitted values are **SNA**, **LZW**, or the name of a compression script, followed by any necessary parameters. The default is **LZW**.

CONDITION

Defines a condition that must be met in order for the change file to be processed. You can specify a condition in relation to:

- The hardware present at a workstation
- The software installed at a workstation
- The operating system running at a workstation

Conditions can be specified in the main section of a profile, or in the dynamic section of a profile (see “Dynamic Section Keywords” on page 264). If it is located in the main section, then it affects the entire change file. If it is located in a dynamic section, then it affects only the objects that call the dynamic section.

You can express either a simple condition or a composite condition. An entire condition can have up to 4096 characters. The syntax for a simple condition is:

CONDITION: <prefix> <token> <operator> <value>

Where:

<prefix>

Specifies the type of condition you are defining. It can be one of the following:

THW A hardware parameter defined by the user when a target was defined to the TME 10 Software Distribution system. The value allowed in the <token> field must be one of those defined by the user.

For example:

CONDITION: THW.DISK_SPACE>120M

Specifies that the change file is to be processed only on workstations with more hard disks that have more than 120-megabytes. The DISK_SPACE have parameter must have been defined for the target.

- TPM** An installation parameter defined at a target. Installation parameters usually specify directory paths to be used when an installation is performed.
- DHW** A discovered hardware parameter. These parameters are detected by the inventory discovery procedure when it is run for a target.
- OPS** The operating system running at the target. The token must be OPERATING_SYSTEM. The value can be:
- AIX
 - HP_UX
 - OS/2
 - WINDOWS (for Windows 3.11)
 - WIN95 (for Windows 95)
 - WIN98 (for Windows 98)
 - WIN2K (for Windows 2000)
 - WINDOWS_NT (for Windows NT)
 - DOS
 - SOLARIS
 - SUNOS
 - NCR
 - MAC
 - SCO
 - SINIX
 - NETWARE
 - IRIX

For example:

```
CONDITION: OPS.OPERATING_SYSTEM = AIX
```

<token>

The token that describes the subject of the condition. The name you provide must be compatible with the prefix, and must be defined to the system as either a hardware parameter, a discovered hardware parameter, an installation parameter or an operating system.

Tokens can also be expressed as:

```
prefix.token_value/subtoken_value
```

This form is used for discovered hardware parameters (DHW prefix) that must be managed in the same way at a target, even if the name of the token and the subtoken is different. For more information see the documentation on inventory discovery.

<operator>

The condition's operator. It can be one of the following:

- = Equal to
- != Different than
- < Less than
- > Greater than
- <= Less than or equal to

>= Greater than or equal to

<value>

The value to apply to the condition. If the value is alpha-numeric, only the = or != operators are allowed.

The syntax for a composite condition is:

CONDITION: <simple_condition> <negation/combination operator> <simple_condition> ►
(optional negation/combination operator)

Where:

<negation/combination operator>

Is an optional operator that links the simple conditions. It can be one of the following:

- blank** When this is the first condition or the only condition, and it's value must be true
- & (AND)** When this is not the first condition in a composite condition, and both this condition and the previous condition must be true
- | (OR)** When this is not the first condition in a composite condition, and either this condition or the previous condition must be true
- ! (NOT)** When this is the first condition or the only condition, and it's value must be false
- &! (AND NOT)** When this is not the first condition in a composite condition, and this condition must be false
- !| (OR NOT)** When this is not the first condition in a composite condition, and either this condition and the previous condition must be false

(An optional open parentheses.

<negation operator>

An optional negation/combination operator.

) Optional closing parentheses.

DATA ACCESS KEY

The data access key associated with the change file. It can have a maximum of eight alphanumeric characters except for *, ?, and /. This keyword is optional. If you do not specify a data access key, no key is assigned with the file. If you do specify a data access key, it must exist and it must be associated with the user who issues the build command for the change file.

REMOVABLE

Specifies whether the change file can be installed as removable. Specify one of the following values:

- YES** The change file can be removed, and all installation options can be specified. This is the default.
- NO** The change file cannot be installed as removable. If a removable installation is attempted, an error message is logged and the installation is not performed.

Change–File Profile Keywords

ACTIVABLE

Specifies whether the change file can be installed in the service area and activated at a later date. Specify one of the following values:

- YES** The change file can be activated, and all installation options can be specified. This is the default.
- NO** The change file cannot be activated, and therefore can only be installed in the active area. If installation in the service is attempted, an error message is logged and the installation is not performed.

AUTHORIZE

Specifies the authorization mode associated with the change file. Specify one of the following values:

- NONE** No target is authorized to install the change file unless explicitly authorized using the **auth** command. This is the default.
- ALL** All targets are authorized to install the change file unless explicitly unauthorized using the **unauth** command.

SW HISTORY RESET

Specifies that after installation, the target change control history records report this as the only change file installed at the targets addressed. Specify one of the following values:

- YES** The change control history records of all change files previously installed at the targets are reset, and an informational message is logged. This change file is not installed on a target if its change control history includes an inactive change file.
- NO** No particular processing is performed. This is the default.

INTERACTIVE

Specifies that the change file contains an interactive procedure, therefore an 'install immediately' request is required. Specify one of the following values:

- YES** The change file can only be installed immediately. If the opposite occurs, a warning message is issued.
- NO** All installation options can be specified. This is the default.

COST

Specifies the cost of the products included in the change file. Enter the cost using the following syntax:

COST: <integer constant>

If you do not specify this keyword, TME 10 Software Distribution, adds the cost associated with each product included in the change file, and assumes this sum as the change file cost.

INSTALLATION DURATION

Specifies the estimated amount of time for installation to complete. Enter this keyword using the following syntax:

INSTALLATION DURATION: <HH:MM:SS>

The purpose of this keyword is to inform the user of the time required to install the change file.

COMMANDS and SCRIPTS

You can specify the names of commands or scripts to execute before or after change control operations. If the commands or the scripts already exist at the target, you do not have to include them in the change file. If a script does not exist at the target, and therefore you are including the script itself in the change file, you must specify the complete path where the script will be installed at the target.

You can specify either a command or the name of a script for the following keywords:

- **PREREQ COMMAND**
- **POSTREQ COMMAND**

You can run these commands or scripts (including parameters) before and after every change control request.

You can specify the name of a script, including its parameters, for any of the keywords listed below. Specify the full path name of the script on the workstation where the change file is being installed.

- **PRE-INSTALL**
- **POST-INSTALL**
- **PRE-ACCEPT**
- **POST-ACCEPT**
- **PRE-REMOVE**
- **POST-REMOVE**
- **PRE-UNINSTALL**
- **POST-UNINSTALL**
- **PRE-ACTIVATE** (not supported for installp change files)

DISK SPACE CHECK

Disk space required in a particular directory on the target or group of targets where the change file is to be installed. Enter it in the form:

<directory name> <required size in KB>

You must leave at least one space between the directory name and the required size values. You can enter several space checks, referring to different directories.

* This keyword cannot be used in AIX installp change file profiles.

SOFTWARE PREREQUISITE

Global file name of the prerequisite change file, which must be already installed. You can enter several software prerequisites.

HARDWARE PREREQUISITE

Hardware prerequisite that must be present before the change file can be installed. It can have one of the following forms:

- *name = value*
- *name ! = value*
- *name < value*

Change-File Profile Keywords

- *name > value*
- *name < = value*
- *name > = value*

You must enter a blank space before and after the = operator. For example, enter !=, not !=. The syntax for the parameter is thus:

```
<name><space><operator><space><value>
```

You can enter more than one hardware prerequisite.

Hardware prerequisites can also be expressed as conditions (see the description of the CONDITION keyword). For example, the prerequisites:

```
HARDWARE PREREQUISITE: DRIVE_SIZE >= 120M
```

```
HARDWARE PREREQUISITE: PROCESSOR >= 386
```

Could also be expressed as:

```
CONDITION: (DHW.DRIVE_SIZE >= 120M ) & (THW.PROCESSOR >= 386 )
```

In both cases, the change file will be installed only if the target has a drive with more than 120 MB and a processor that is at least a 386.

DEFAULT TOKEN

Token to be defined during installation of the change file, if not already defined. This takes the form:

```
token=value
```

You can also define tokens to be used on specific operating systems. This takes the form:

```
token(operating_system)=value
```

Where operating_system can be one of the following:

- AIX
- HP_UX
- OS/2
- WINDOWS (for Windows 3.11)
- WIN95 (for Windows 95)
- WIN98 (for Windows 98)
- WIN2K (for Windows 2000)
- WINDOWS_NT (for Windows NT)
- DOS
- SOLARIS
- SUNOS
- NCR
- MAC
- SCO
- SINIX
- NETWARE
- IRIX

You can enter several default tokens.

SECURE PACKAGE

Specify **SECURE PACKAGE** when you want to make sure that the files in the package are the same as those inserted in it when the change file was built. If the names of the files include wildcard characters, then the files must be accessible at build time, because the secure package check is always performed in these cases. Possible values are:

YES When the change file is installed, a check is performed to ensure that the remote files specified in it are accessible, and that they are the same as those inserted in the change file when it was built.

NO No check on remote files is performed when the change file is installed.

The default is **NO**.

REBOOT REQUIRED

Specify **REBOOT REQUIRED** when the target workstation must be restarted in order to activate the changes installed in the active area. Possible values are:

YES A reboot is required.

NO A reboot is not required.

Product Section Keywords

The keywords that can be inserted in the sections of change file profiles that describe products are listed here. In keeping with the POSIX 7.2 definition, a product is “a change file used to define a set of related software. It defines attributes describing the operating systems and the hardware architectures the product supports. These attributes apply to the filesets contained in the product.”

You can specify up to xx **PRODUCT** sections and yy **FILESET** sections in a profile. They are both optional.

PRODUCT

This keyword indicates the start of a product section. No value can be specified for it.

TAG

Specifies a short name associated with the product. This keyword is required. It can be up to 144 characters long.

REVISION

Specifies the revision level of the product. Specify a series of integers separated by periods. This keyword is optional. It can be up to 32 characters long.

ARCHITECTURE

Specifies the hardware requirements for the system. This keyword is optional. It can be up to 32 characters long.

VENDOR TAG

The name of the vendor that supplied the product. This keyword is optional. It can be up to 32 characters long.

Change–File Profile Keywords

TITLE

The name of the product. This keyword is optional. It can be up to 80 characters long.

DESCRIPTION

A more detailed description of the product. This keyword is optional. It can be up to 8192 characters long. The description can be specified in one of the following ways:

- Enter multiple lines in the profile. Enclose the descriptions between double quotation marks. Within the description itself, enter double quotation marks as \“.
- Enter < followed by the name of a file that contains the description. The contents of the file are copied into the profile.

COST

The cost of the product. This is not a POSIX 7.2 keyword.

FILESET

Indicates a FILESET section. In keeping with POSIX 7.2, a fileset is a “change file used for grouping a set of software files into a manageable object. The fileset is the smallest level of software that can be managed by the tasks defined in the POSIX standard.”

At least one fileset section must be specified for a product. It can include the following keywords:

TAG

Specifies a short name associated with the fileset. This keyword is required. It can be up to 144 characters long.

REVISION

Specifies the revision level of the fileset. Specify a series of integers separated by dots. This keyword is optional. It can be up to 32 characters long.

TITLE

The name of the fileset. This keyword is optional. It can be up to 80 characters long.

Remote Directory Section Keywords

The keywords that can be inserted in the sections of change file profiles for remote directories are described here.

REMOTE DIRECTORY

Specifies the remote directories that contain the change file contents and that must be accessed to perform the installation process. The export and mount parameters for a list of up to 16 directories can be specified. A remote directory section looks like this:

```
REMOTE DIRECTORY:
EXPORTED DIRECTORY: /velka/source/aix/bin
MOUNTED FILE SYSTEM: /home/r2build/bin
MOUNT OPTIONS: -b 4092
EXPORT OPTIONS: -ro
```

You can specify the following keywords:

EXPORTED DIRECTORY

The name of the remote directory to mount. It is required.

MOUNTED FILE SYSTEM

The name of the file system from which the exported directory is to be mounted. It is required.

SERVER NAME

The server where the exported directory is located. If this keyword is not specified, the local server is used.

MOUNT OPTIONS

The options that can be specified for the mount command.

EXPORT OPTIONS

The options that can be specified for the export command. This keyword is valid only if SERVER NAME indicates the local server.

Object Section Keywords

An object section must begin with the OBJECT: keyword, which has no value specified. It must be the last keyword entered in the command. You can specify more than one object in a profile. An object section looks like this:

```
OBJECT:
SOURCE NAME:      /u/user/dir2
TARGET NAME:      /u/user/dir2
TYPE:             DIRECTORY
ACTION:           CREATE
```

The keywords that can be inserted in the object sections of change file profiles are:

DYNAMIC SECTION

The name of the dynamic section whose conditions must be satisfied in order for the object to be processed at a target. See “Dynamic Section Keywords” on page 264.

You can specify more than one dynamic section keyword for an object. If you do, the object is processed if at least one of the conditions in the called dynamic sections are met.

The dynamic section or sections specified with this keyword must exist in the change file profile, otherwise the change file build fails.

SOURCE NAME (m)

Source name of files to be included in the change file. Use this name to locate the files when building the change file. Specify the path where the files are stored. If you are specifying files that reside at a remote site, specify the path at the build

Change-File Profile Keywords

site that accesses the path at remote site. The name must begin with the / character.

The file name can contain wildcards so that more than one file is included in the change file. One **SOURCE NAME** must be specified for every object. This keyword is ignored for objects of TYPE=DIRECTORY or objects with ACTION=DELETE.

* In AIX installp change file profiles, **SOURCE NAME** is replaced by the **NAME** keyword. The **NAME** can refer to an installp image or to script files. The name cannot be a directory, but you can use wildcards to specify more than one installp image.

SOURCE NAME AT INSTALL

Use this name to locate remote files at a remote source when building a change file using files that reside at remote sites. It can only be used if the object **TYPE** is:

- REMOTE_FILE
- REMOTE_FILE_WITH_TOKENS
- REMOTE_IMAGE
- REMOTE_IMAGE_PMP

Specify the path at the target that accesses the path at the remote site where the files are stored. The name must begin with the / character or a token. The file name can contain wildcards so that more than one file is included in the change file. One **SOURCE NAME AT INSTALL** must be specified for every object.

If this keyword is not specified, it defaults to the SOURCE NAME.

* In AIX installp change file profiles, **SOURCE NAME AT INSTALL** is replaced by the **REMOTE NAME AT INSTALL** keyword. When **REMOTE NAME AT INSTALL** is used, the same directory path must be used for all the images in the change file. In addition, a change file must contain either all remote or all local images, they cannot contain a mixture of the two.

OPP OPTION

* Use this keyword *only* in AIX installp change profiles to specify which OPP options, contained within the AIX installp image, are to be installed. The OPP option keyword must include the option name *and* version information, so that the precise installp image to be installed is identified. If the version is omitted, an error occurs during the change file build process. The AIX **installp -l** command can be used to obtain the option name and version information required.

Multiple OPP OPTION keywords are allowed. If all options for a specific installp image are to be installed, enter the product name followed by **.all** (for example, bosnet.all).

This keyword is not valid if the name of the object contains wildcard characters or if the name refers to a directory. In these cases, *all* options in the images are installed.

If the OPP OPTION keyword is not specified, all options are included in the change file.

TARGET NAME

The path and name that the files specified by this object will have when the change file is installed at the target. The name must begin with the / character or a token. Tokens are expanded by the target at the time of installation. If this keyword is not included, the target name defaults to the source name. If you do not specify a directory name, the file is installed into the same directory as the previous file in the change file (the previous object).

TYPE

Object type. You must include a type for every object. Possible values are:

FILE

Used for generic files. Use FILE instead of INSTALLP_IMAGE if the **NAME** keyword refers to an image name containing wildcards.

FILE_WITH_TOKENS

Used for generic files. When this type is specified, any tokens within the files in the object are expanded at the time of installation.

REMOTE_FILE

Used for remote files that do not contain tokens. The object header contains the file attributes, but the file itself is not included in the change file.

REMOTE_FILE_WITH_TOKENS

Used for remote files that contain tokens. The object header contains the file attributes, but the file itself is not included in the change file. Tokens are translated when a change file is installed.

DIRECTORY

To create a directory, the directory must exist on the preparation site workstation. Its attributes are copied into the change file. Specify the name of the directory in the **TARGET NAME** keyword in the change file profile.

Directories can be specified if the keyword ACTION has either one of the following values:

- Create
- Delete

INSTALLP_IMAGE*

* Used for AIX installp files. INSTALLP_IMAGES cannot be associated with ACTIONS.

INSTALL_IMAGE_PMP

* Used for AIX installp preventive maintenance packages (PMPs). The /usr/sys/inst.images is used to access the PMP images during installation. This directory must be mounted before the installation begins. INSTALLP_IMAGE_PMPs cannot be associated with ACTIONS.

REMOTE_IMAGE

* Used for AIX installp files that reside at a remote site. All installp images included in a change file must be either remote or local, a mixture of the two cannot be used. REMOTE_IMAGES cannot be associated with ACTIONS.

REMOTE_IMAGE_PMP*

* Used for AIX installp preventive maintenance packages (PMPs) that reside at a remote site. The directory name to be used during installation is taken from the REMOTE NAME keyword if it is specified, or if not from the NAME keyword. REMOTE_IMAGE_PMPs cannot be associated with ACTIONS.

ACTION

Action to be taken when installing the object.

COPY

Currently, you can use this action for the following object TYPES:

- FILE
- FILE_WITH_TOKENS
- REMOTE_FILE
- REMOTE_FILE_WITH_TOKENS

CREATE

Use this action only for DIRECTORY object types. It creates a directory with the current date and time. The directory does not have to exist at the workstation where the change file is built. If parent directories do not exist, they are created. If the directory exists, its attributes are not changed and a warning message is issued.

DELETE

Delete an object.

Use this action for the following object TYPES:

- FILE

If the change file is installed as removable, when it is deleted a backup is performed; it can be restored by removing the change file. If the change file is uninstalled, it remains deleted.

- DIRECTORY

When a directory is deleted, its contents and the contents of its subdirectories are deleted. If the change file is installed as removable, when the directory is deleted a backup is performed; the directory can be restored by removing the change file. If the change file is uninstalled, the directory and its contents remain deleted, and backups are deleted as well.

OWNER

The ID of the user at the workstation where the change file is built. This keyword is optional, and is only valid for generic change files. You cannot use it when the ACTION specified is DELETE, or when the TYPE specified is REMOTE_FILE or REMOTE_FILE_WITH_TOKENS. For FILE or FILE_WITH_TOKENS, only the root user can change this keyword. When a Generic change file is built on a non-UNIX system, owner defaults to root or FNDADMN.

If the OWNER specified does not exist, the default owner (ROOT) is used.

GROUP

The name of the group of files this file belongs to. This keyword is optional, and is only valid for Generic change files. You cannot use it when the ACTION specified is DELETE, or when the TYPE specified is REMOTE_FILE or REMOTE_FILE_WITH_TOKENS. For FILE or FILE_WITH_TOKENS, this keyword can only be changed by the current owner of the file or by the root user.

If the GROUP specified does not exist at a target, the default group (system) is used.

GENERAL ATTR

The attributes specified for a DOS, OS/2, or NetWare change file. You cannot use this keyword when the ACTION specified is DELETE, or when the TYPE specified is REMOTE_FILE or REMOTE_FILE_WITH_TOKENS. For FILE or FILE_WITH_TOKENS, this keyword can be changed by any user.

General attributes are as follows:

```
<attributes> = [-|R] [-|A] [-|H]
               [-|S] [-|X] [-|B]
```

where attribute is one of the following:

R= Read only
 A= Archive
 H= Hidden
 S= System
 X= EXecute only
 B= Shareable

UNIX ATTR

The attributes specified for a UNIX change file. You cannot use this keyword when the ACTION specified is DELETE, or when the TYPE specified is REMOTE_FILE or REMOTE_FILE_WITH_TOKENS. For FILE or FILE_WITH_TOKENS, this keyword can only be changed by the current owner of the file or by the root user.

UNIX attributes are as follows:

```
<attributes> = [-|d] [-|r] [-|w]
               [-|x|s|S] [-|r] [-|w]
               [-|x|s|S] [-|r] [-|w]
               [-|x|t|T]
```

NETWARE ATTR

The attributes specified for a NetWare change file. You cannot use this keyword when the ACTION specified is DELETE, or when the TYPE specified is REMOTE_FILE or REMOTE_FILE_WITH_TOKENS. For FILE or FILE_WITH_TOKENS any user can change this keyword.

NetWare attributes are as follows:

```
<attributes> = [-|T] [-|R] [-|W]
               [-|P] [-|N] [-|D]
               [-|C]
```

Dynamic Section Keywords

Where attribute can be one of the following:

T= Transaction bit
R= Read audit
W= Write audit
P= Immediate Purge
N= ReName inhibit
D= Delete inhibit
C= Copy inhibit

INCLUDE SUBDIRS

Specifies whether to search subdirectories for files matching the file specification in the source name. Allowed values are **YES** and **NO**. The default is **NO**. This keyword is ignored for objects of TYPE=DIRECTORY or objects with ACTION=DELETE.

Dynamic Section Keywords

You call dynamic sections from object sections of a profile, according to your operating needs. Dynamic sections can contain information pertaining to:

- Software prerequisites
- Remote directories that must be mounted for an installation
- Commands or procedures to be executed at targets before or after a change control operation

For example, you can use dynamic sections to create a change file that can be installed on two types of workstation: small machines or big machines. To do so you would create a profile with these sections:

```
DYNAMIC SECTION:      big_machine
  CONDITION:           DRIVE_SIZE >= 180M
  PRE-INSTALL:         /usr/lpp/netviewdm/pre-install

DYNAMIC SECTION:      small_machine
  CONDITION:           DRIVE_SIZE < 180M
  PRE-INSTALL:         /usr/lpp/netviewdm/pre-install-small

OBJECT:
  DYNAMIC SECTION:     small_machine
  SOURCE NAME:         user/lpp/netviewdm/file_small
  TYPE:                FILE
  ACTION:              COPY
```

If you specify conditions in the envelope section of a change file profile that includes dynamic sections as well, when an operation is requested for a target it will be executed only if the conditions specified at the envelope level are met. Files in object sections that include calls to dynamic sections will be processed only if the condition in the dynamic section is met as well.

A dynamic section that contains only a `CONDITION` and or a `SOFTWARE PREREQUISITE` keyword, *must* be called by an object section in the same profile, or the change file build fails.

Dynamic sections must begin with a unique `DYNAMIC SECTION NAME`. It assigns a name which is used in object sections (using the `DYNAMIC SECTION` keyword) to call the dynamic section.

Dynamic sections can contain these keywords:

`CONDITION`

The section's condition, which is expressed using the format described for the `CONDITION` keyword on page 251. A condition must be specified.

`SOFTWARE PREREQUISITE`

An optional software prerequisite, expressed in the format described in 252.

`REMOTE DIRECTORY`

An optional statement that defines the remote location where software is stored, as described in "Remote Directory Section Keywords" on page 258.

`PRE-, POST- ACTIONS`

Optional statements that define procedures or commands to be executed before or after the change control operation, as described on page 255.

Examples of TME 10 Software Distribution Change File Profiles

This change file profile contains two fixes to the change file that updated level 2 to level 3 of the Eurotravel word processor.

Examples of TME 10 Software Distribution Change File Profiles

```
GLOBAL NAME:          EURO.WORDPROC.FIX.3.F0002.US
DESCRIPTION:          Fix to bugs F0001 and F0002
CHANGE FILE TYPE:     GENERIC
COMPRESSION TYPE:     LZW
SECURE PACKAGE        YES
REBOOT REQUIRED        YES

SOFTWARE PREREQUISITE: EURO.WORDPROC.UPD.2.3.US

HARDWARE PREREQUISITE: mem > 6000
HARDWARE PREREQUISITE: monitor = SVGA

OBJECT:
  SOURCE NAME:        /usr/wordproc/fix/*
  TARGET NAME:        $(EuroWord)/*
  TYPE:               FILE
  ACTION:             COPY

OBJECT:
  SOURCE NAME:        /usr/wordproc/menus
  TARGET NAME:        $(EuroWord)/menus
  TYPE:               FILE
  ACTION:             COPY

OBJECT:
  SOURCE NAME:        /u/user/dir2
  TARGET NAME:        /u/user/dir2
  TYPE:               DIRECTORY
  ACTION:             CREATE

OBJECT:
  SOURCE NAME:        /u/nfs/remote_file
  TARGET NAME:        /u/user/remote_file
  TYPE:               REMOTE FILE
  ACTION:             COPY
```

Figure 2. Example of a TME 10 Software Distribution Change File profile

The software prerequisite was specified because this fix applies only if the update from level 2 to level 3 has been used. If level 3 was attained by installing the change file containing the refresh version of level 3, this fix is not appropriate.

The first object stores all of the files from `/usr/wordproc/fix` when the change file is built, and copies them into the directory specified by the token `EuroWord` when the change file is installed.

The second object stores the file called `/usr/wordproc/menus` when the change file is built, and copies it into the directory specified by the token `EuroWord` when the change file is installed.

The following is an example of a change file profile for TME 10 Software Distribution.

```

GLOBAL NAME:      NETWORK.PLAN.REF.3
DESCRIPTION:      Network planner 3.1
CHANGE FILE TYPE: change file type
COMPRESSION TYPE: LZW
REMOVABLE:        NO
ACTIVABLE:        YES
SW HISTORY RESET: YES
AUTHORIZE:        ALL
INTERACTIVE:      YES
COST:             1500
INSTALLATION TIME: 4

PRODUCT:
TAG:              NETPLAN
REVISION:         3.1
ARCHITECTURE:     platform architecture
VENDOR TAG:       IBM
TITLE:            Network Planner
DESCRIPTION:      < readme.txt
COST:             1000

FILESET:
TAG:              Base
REVISION:         3.1.1
TITLE:            Base feature

FILESET:
TAG:              Samples
REVISION:         3.1.5
TITLE:            Sample feature

SOFTWARE PREREQUISITE: NET.2.3.US

HARDWARE PREREQUISITE: mem > 6000
HARDWARE PREREQUISITE: monitor = SVGA

REMOTE DIRECTORY:
EXPORTED DIRECTORY: /usr/lpp/netviewdm
MOUNTED FILE SYSTEM: /home/source
SERVER NAME:        srv03
MOUNT OPTIONS:      -b 4092
EXPORT OPTIONS:     -ro

REMOTE DIRECTORY:
EXPORTED DIRECTORY: /velka/source/aix/bin
MOUNTED FILE SYSTEM: /home/r2build/bin
MOUNT OPTIONS:      -b 4092
EXPORT OPTIONS:     -ro

OBJECT:
REMOTE SOURCE NAME: /home/source/*
TARGET NAME:        $(NetPlan)/*
TYPE:               REMOTE FILE
ACTION:             COPY

OBJECT:
REMOTE SOURCE NAME: /home/r2build/bin
TARGET NAME:        $(NetPlan)/menus
TYPE:               REMOTE FILE
ACTION:             COPY

```

Figure 3. Example of a TME 10 Software Distribution Change File profile

Example of an AIX installp Change File Profile

This example shows an AIX installp change file profile.

Example of an AIX installp Change File Profile

```
GLOBAL NAME:          EURO.PRODUCT.REF.1
DESCRIPTION:          First product
LOCAL NAME:           /usr/sys/inst.images/euro.prod
CHANGE FILE TYPE:     AIXINSTP
COMPRESSION TYPE:     LZW
PACK FILES:           NO
OBJECT:
  NAME:               /images/euro.prod.1
  OPP OPTION:         euro.obj 01.00.0000.0000
  TYPE:               INSTALLP_IMAGE
```

Figure 4. AIX installp Change File profile

Example of AIX installp Change File Profile Objects

Figure 5 shows examples of different objects defined for an installp change file profile.

```
OBJECT:
  NAME:               /images/base/bsmEn_US
  OPP-OPTION:         bsmEn_US.msg 03.02.0000.0000
  TYPE:               INSTALLP_IMAGE
OBJECT:
  NAME:               /images/base/INed.usr.3.2.0.0
  OPP OPTION:         INed.obj 03.02.0000.0000
  TYPE:               INSTALLP_IMAGE
OBJECT:
  NAME:               /images/base/pci
  TYPE:               INSTALLP_IMAGE
OBJECT:
  NAME:               /images/base/txtfmt.usr.3.2.0.0
  OPP OPTION:         txtfmt.tfs.obj 03.02.0000.0000
  OPP OPTION:         txtfmt.xpv.obj 03.02.0000.0000
  OPP OPTION:         txtfmt.spell.obj 03.02.0000.0000
  TYPE:               INSTALLP_IMAGE
OBJECT:
  NAME:               /images/updates/*
  TYPE:               FILE
```

Figure 5. AIX installp Change File profile objects

Note: You can use any names for the image file names when storing the file, but the OPP option name must be the actual name displayed from the installp -l command.

Example of AIX installp -l Output

Figure 6 displays an example of the output from the AIX installp -l command.

Example of AIX installp -l output for the /images/base directory			
Option Name	Level	I/U Q Content	
=====			
bsmEn_US.msg	03.02.0000.0000	I	N usr
# Base System Messages - U.S. English			
INed.obj	03.02.0000.0000	I	N usr
# INed Editor			
pci.obj	03.02.0000.0000	I	N usr
# AIX DOS Server for use with AIX Access for DOS Users			
txtfmt.tfs.obj	03.02.0000.0000	I	N usr
# Text Formatting Services			
txtfmt.spell.obj	03.02.0000.0000	I	N usr
# Writer's Tools			
txtfmt.bib.obj	03.02.0000.0000	I	N usr
# Bibliography Support			
txtfmt.xpv.obj	03.02.0000.0000	I	N usr
# Troff Xpreviewer			
txtfmt.ts.obj	03.02.0000.0000	I	N share
# TranScript Tools			
txtfmt.graf.obj	03.02.0000.0000	I	N share
# Base Graphic Commands			
txtfmt.hplj.obj	03.02.0000.0000	I	N share
# Printer Font Support			
txtfmt.ibm3812.fnt	03.02.0000.0000	I	N share
# Printer Font Support			
txtfmt.ibm3816.fnt	03.02.0000.0000	I	N share
# Printer Font Support			

Figure 6. Example of AIX installp -l output

Example of an AIX installp Change File Profile

Index

Special Characters

- o, connect command 57
- r, connect command 56
- r, disconnect command 75
- s, connect command 56
- t, telephone number 56

A

- acc command 22
- access keys
- act command 27
- active, change file status 123
- addgp command 31
- addpln command 33
- addpm command 34
- addprf command 36
- addtg command 39
- addusr command 45
- auth command 47
- authorizations
 - addusr command 45
 - auth command 47
 - command-line interface 18
 - lsusr command 141
 - unauth command 202
- available, change file status 122

B

- back level, change file status 122
- bld command 51

C

- cat command 52
- catalogs
 - cat command 52
 - ls command 114
 - lsak command 116
 - uncat command 209
- change file 122
- change file profiles
 - creating 247
 - dynamic 247

- change file profiles (*continued*)

- envelope keywords 250
 - examples 265
 - examples of 265
 - keywords 250
 - static 247
 - structure 247
 - syntax 247

- change files

- acc command 22
 - active status 123
 - available, status 122
 - back level, status 122
 - bld command 51
 - building change files 51
 - creating profiles 247
 - del command 58
 - delcm command 62
 - discovered, status 122
 - distributed, status 122
 - distribution pending, status 122
 - example of profile 265
 - imp command 97
 - in error status 122
 - inactive status 123
 - inst command 99
 - installed status 122
 - lscf command 118
 - lscm command 121
 - not available status 122
 - not removable status 123
 - reboot required, status 123
 - rem command 159
 - removable status 123
 - rtv command 169
 - scheduled, status 122
 - send command 179
 - unbld command 206
 - unpacking change files 206
 - updcn command 218
 - vercm command 241

- change management

- acc command 22
 - act command 27
 - delprf command 70
 - delrq command 71

change management (*continued*)

- lsrq command 132
- rem command 159
- uninst command 210
- vercm command 241

clients

command-line interface 1

- acc 22
- accepting change files 22
- act 27
- activating changes on targets 27
- addgp 31
- adding parameters to targets 34
- adding plans 33
- adding target groups 31
- adding user profiles 36
- adding users 45
- addpln 33
- addpm 34
- addprf 36
- addtg 39
- addusr 45
- authorizations 18
- authorizing targets to access files 47
- bld 51
- cat 52
- checking communication status 190
- closing command line session 152
- connect 56
- connect to server from mobile client 56
- copying files from servers 89
- creating catalog entries 52
- creating prerequisite plans 145
- creating targets 39
- dates 13
- del 58
- delcm 62
- delete local files 58
- deleting authorization profiles 70
- deleting change file status 62
- deleting parameters from targets 69
- deleting requests 71
- deleting target groups 67
- deleting targets 73
- deleting users 74
- delgp 67
- delpm 69
- delprf 70
- delrq 71
- deltg 73

command-line interface (*continued*)

- delusr 74
- disconnect 75
- disconnect mobile client connection 75
- displaying message log 111
- displaying queue entry details 129
- entering data 11
- eraserq 76
- erasing requests 76
- examples of date and time usage 15
- exec 78
- execf 82
- execpln 86
- executing noncataloged programs or scripts 82
- executing plans 86
- executing programs or scripts 78
- exp 89
- giving priority to a queue 150
- hdlrq 94
- help 90
- hldc 91
- hldq 92
- hldtg 96
- holding queues 92
- holding remote communications 91
- holding requests 94
- holding targets 96
- imp 97
- importing change files 97
- inst 99
- installing change files 99
- inv 106
- listing access keys 116
- listing authorization profiles 127
- listing catalog entries 114
- listing change file contents 118
- listing plan contents 126
- listing request status 132
- listing target groups 124
- listing target status 193
- listing targets 137
- listing user authorizations 141
- log 111
- ls 114
- lsak 116
- lsbs 117
- lscf 118
- lscm 121
- lsgp 124
- lspln 126

- lsprf 127
- lsq 129
- lsrq 132
- lstg 137
- lsusr 141
- messages 17
- nvdn 143
- open 144
- opening communication to server 144
- performing inventory discovery 106
- preqln 145
- prgq 148
- procedures after requests 16
- prtyq 150
- purging entries 148
- quit 152
- recursive parameters 16
- relc 153
- releasing queues 154
- releasing requests 156
- releasing SNA communication 153
- releasing targets 158
- reloading configuration 167
- relq 154
- relrq 156
- reltg 158
- rem 159
- removing catalog entries 209
- removing change files 159
- removing target authorizations 202
- renaming targets 164
- rentg 164
- reporting status of files 121
- reset 166
- resetting TME 10 Software Distribution 166
- restarting requests 168
- retrieving cataloged files 169
- retrieving noncataloged files 174
- return codes 17
- rld 167
- rstrq 168
- rtv 169
- rtrvf 174
- selecting servers 197
- send 179
- sendf 184
- sending cataloged files 179
- sending noncataloged files 184
- setting internal tracing 198

- starting command line session 143
- starting traces 201
- stat 190
- stattg 193
- stop 196
- stopping traces 200
- summary of commands 1
- svr 197
- syntax rules 9
- times 13
- trace 198
- troff 200
- tron 201
- unauth 202
- unbld 206
- uncat 209
- uninst 210
- uninstalling files 210
- upd 215
- updak 216
- update change control status 218
- updating access keys 216
- updating authorization profiles 224
- updating catalog entries 215
- updating configurations 217
- updating plans 223
- updating requests 228
- updating targets 232
- updating user authorizations 239
- updating user passwords 227
- uppbs 217
- updcn 218
- updpln 223
- updpfr 224
- updpwd 227
- updrq 228
- updtg 232
- updsr command 239
- vercm 241
- verifying change file status 241
- command-line operations 1
- conditions, dynamic change files 253
- connect command 56
- creating change file profiles 247

```
data files
del command 58
```

data files (*continued*)

- rtv command 169
- rtvf command 174
- send command 179
- sendf command 184
- del command 58
- delete pending request xiii
- delf command 63
- delgp command 67
- delpm command 69
- delprf command 70
- delrq command 71, 74
- deltg command 73
- disconnect command 75
- discovered, change file status 122
- distributed, change file status 122
- distribution
 - lsq command 129
 - lsrq command 132
 - prgq command 148
 - rtv command 169
 - rtvf command 174
 - send command 179
 - sendf command 184
- distribution pending, change file status 122
- duration, connection window 56
- dynamic change file 247
- dynamic section, change file profile 264

E

- envelope keywords, change file profile 250
- eraserq command 76
- exec command 78
- execf command 82
- execpln command 86, 94
- exp command 89

G

- global name 13

H

- help command 90
- hldc command 91
- hldq command 92
- hldtg command 96
- host deletion of pending requests xiii

I

- imp command 97
- in error, change file status 122
- inactive, change file status 123
- inst command 99
- installed, change file status 122
- installp change files
 - change file profile keywords 250
 - creating profiles 247
- inv command 106
- inventory discovery 106

L

- language variable 17
- language variable, 17
- local target statuses, changes to xiv
- log command 111
- ls command 114
- lsak command 116
- lsbs command 117
- lscf command 118
- lscm command 121
- lsgp command 124
- lspln command 126
- lsprf command 127
- lsq command 129
- lsrq command 132
- lstg command 137
- lsusr command 141

M

- messages
 - command-line interface 17
 - log command 111
 - troff command 200
 - tron command 201

N

- new in this release xi
- not available, change file status 122
- not removable, change file status 123
- nvdn command 143

O

open command 144
outstanding request, connection window 57

P

parameters, recursive 16
pending requests, delete xiii
platforms supported xi
pregpln command 145
prgq command 148
pristine installation scenarios, new xi
profiles, change file 247
 creating 247
 examples 265
 keywords 250
prtyq command 150

Q

queues
 delrq command 71
 eraserq command 76
 hldc command 91
 hldq command 92
 lsq command 129
 lsrq command 132
 prgq command 148
 prtyq command 150
 relc command 153
 relq command 154
quit command 152

R

reboot required, change file status 123
recursion, connection window 56, 75
recursive parameters 16
recursive requests
 relc command 153
 relq command 154
 relrq command 156
 reltg command 158
rem command 159
removable, change file status 123
rentg command 164
reset command 166
return codes, command-line interface 17

rld command 167
rstrq command 168
rtrv command 169

S

scheduled, change file status 122
scripts
 exec command 78
 execf command 82
send command 174, 179, 184
server
 exp command 89
 svr command 197
start command 189
start time, connection window 56
stat command 190
stattg command 193
stattg command, changes to statuses xiv
status 122
 active, change file 123
 available, change file 122
 back level, change file 122
 delcm command 62
 discovered, change file 122
 distributed, change file 122
 distribution pending, change file 122
 in error, change file 122
 inactive, change file 123
 installed, change file 122
 lscm command 121
 not available, change file 122
 not removable, change file 123
 reboot required, change file 123
 removable, change file 123
 scheduled, change file 122
 stat command 190
stop command 196
svr command 197
syntax rules, command-line interface 9

T

target name, connect command 56
targets
 act command 27
 addgp command 31
 addpm command 34
 addtg command 39
 auth command 47

targets (*continued*)

- delgp command 67
- delpm command 69
- deltg command 73
- entering names, command-line interface 12
- exec command 78
- execf command 82
- hldtg command 96
- lsak command 116
- lsgp command 124
- lstg command 137
- reltg command 158
- rem command 159
- rentg command 164
- stattg command 193
- unauth command 202
- updtg command 232

telephone number, connect command 56

time, command-line interface 13

TME 10 Software Distribution

- reset command 166
- start command 189
- stop command 196

trace command 198

transmission plans

- addpln command 33
- deleting requests 71
- eraserq command 76
- execpln command 86
- hldrq command 94
- lspln command 126
- lsrq command 132
- pregpln command 145
- relrq command 156
- rstrq command 168
- updpIn command 223
- updrq command 228

troff command 200

tron command 201

U

- unauth command 202
- unbld command 206
- uncat command 209
- uninst command 210
- updak command 216
- updbS command 215, 217
- updcM command 218

- updpIn command 223
- updpri command 224
- updpwd command 227
- updrq command 228
- updtg command 232
- updusr command 239

users

- addprf command 36
- addusr command 45
- delpri command 70
- delusr command 74
- lsprf command 127
- lsusr command 141
- updak command 216
- updpri command 224
- updpwd command 227
- updusr command 239

V

- vercm command 241

W

- what's new xi

Communicating Your Comments to IBM

TME® 10 Software Distribution
for AIX®
Command Reference
Version 3.1.5
Publication No. xxxx-xxxx-xx

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you prefer to send comments by mail, use the reader's comment form (RCF) at the back of this book. If you wish, you can give the RCF to the local branch office or IBM representative for postage-paid mailing.

If you prefer to send comments by fax, use this number, which is in Italy:

39+06+596+62077

If you prefer to send comments electronically, use this network ID:

ROMERCF at VNET.IBM.COM

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies

Help us help you!

**TME® 10 Software Distribution
for AIX®
Command Reference
Version 3.1.5
Publication No. xxxx-xxxx-xx**

We hope you find this publication useful, readable, and technically accurate, but only you can tell us! Your comments and suggestions will help us improve our technical publications. Please take a few minutes to let us know what you think by completing this form.

Overall, how satisfied are you with the information in this book?	Satisfied	Dissatisfied
	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:	Satisfied	Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your task	<input type="checkbox"/>	<input type="checkbox"/>

Specific comments, problems, or suggestions:

Thank you for your response. When you send information to IBM, you grant IBM the right to use or distribute the information without incurring any obligation to you. You of course retain the right to use the information in any way you choose.

Name	Address
Company or Organization	
Phone No.	

Help us help you!
XXXX-XXXX-XX



Cut or Fold
Along Line

Fold and Tape

Please do not staple

Fold and Tape

PLACE
POSTAGE
STAMP
HERE

License Use Management Information Development
Rome Tivoli Laboratory
IBM Italia S.p.A.
Via Sciangai, 53
00144 Rome
Italy

Fold and Tape

Please do not staple

Fold and Tape

XXXX-XXXX-XX

Cut or Fold
Along Line



Distributed Electronically for Customer Printing