



TME® 10 Software Distribution

# Command Reference

*Version 3.1.5*





TME® 10 Software Distribution

# Command Reference

*Version 3.1.5*



# Contents

<b>What's New in This Release</b> . . . . .	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
<b>Chapter 1. Command-Line Operations</b> . . . . .	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
Entering a User ID and Password . . . . .	8
Entering Commands . . . . .	8
Entering Data in the Command-Line Interface . . . . .	10
Entering Target Names . . . . .	11
Entering Global Names . . . . .	12
Entering Dates and Times . . . . .	12
Running a Procedure after a Request Has Completed . . . . .	15
Return Codes and Error Messages . . . . .	15
Authorization Levels for Command Use . . . . .	16
<b>Chapter 2. TME 10 Software Distribution Line Commands</b> . . . . .	19
acc – Accept an Installed Change File . . . . .	20
act – Activate Changes on a Target . . . . .	25
addgp – Add a Target Group . . . . .	29
addpm – Add Parameters to a Target . . . . .	31
addprf – Add User Profile . . . . .	33
addtg – Create a Target . . . . .	36
addusr – Create a User . . . . .	41
auth – Authorize Targets to Access Files . . . . .	43
bld – Build a Change File on the Server . . . . .	47
cat – Create a Catalog Entry . . . . .	48
connect – Establish Mobile Connection with Server . . . . .	52
del – Delete a File . . . . .	54
delcm – Delete a Change File Status Record . . . . .	58
delf – Delete a Noncataloged File . . . . .	59
delgp – Delete a Target Group . . . . .	63
delpm – Delete Parameters from a Target . . . . .	65
delprf – Delete an Authorization Profile . . . . .	66
delrq – Delete a Request . . . . .	67

deltg – Delete a Target . . . . .	69
delusr – Delete a User . . . . .	70
disconnect – Close Mobile Connection with Server . . . . .	71
eraserq – Erase Requests . . . . .	72
exec – Execute a Program or Script . . . . .	74
execf – Execute a Noncataloged Program or Script . . . . .	78
exp – Copy a File from the Server to Your Workstation . . . . .	82
help – Display Help for the Command-Line Interface . . . . .	84
hldc – Hold Remote Communication . . . . .	85
hldq – Hold One or All Queues . . . . .	86
hldrq – Hold a Request . . . . .	88
hldtg – Hold a Target . . . . .	90
imp – Import a Change File into the System . . . . .	91
inst – Install a Change File . . . . .	92
inv – Inventory Discovery . . . . .	99
log – Display the Contents of the Message Log . . . . .	104
ls – List Entries in the Catalog . . . . .	107
lsak – List Access Keys . . . . .	109
lsbs – List Configuration . . . . .	110
lscf – List the Contents of a Change File . . . . .	111
lscm – Report the Status of Files . . . . .	114
lsgp – List Target Groups . . . . .	117
lspfr – List Profiles . . . . .	119
lsq – Display Queue Entry Details . . . . .	121
lsrq – List Change Control and Distribution Requests . . . . .	124
lstg – List Targets . . . . .	129
lsusr – List User Authorizations . . . . .	133
nvdm – Start a Command Line Session . . . . .	135
open – Open a Connection to a TME 10 Software Distribution Server . . . . .	136
prgq – Purge One or All Entries from a Queue . . . . .	137
prtyq – Give Priority to a Queue . . . . .	139
quit – Close an Interactive Command Line Session . . . . .	141
relc – Release Remote Communication . . . . .	142
relq – Release One or All Queues . . . . .	143
relrq – Release a Request . . . . .	145
reltg – Release a Target . . . . .	147
rem – Remove a Change File From a Target or Group . . . . .	148
rentg – Rename a Target . . . . .	153
reset – Reset TME 10 Software Distribution . . . . .	155
rld – Reload Configuration Information . . . . .	156
rstrq – Restart a Request . . . . .	157
rtrv – Retrieve a File . . . . .	158
rtrvf – Retrieve a Noncataloged File . . . . .	163
send – Send a File to Another Target . . . . .	168
sendf – Send a Noncataloged File . . . . .	173
start – Start TME 10 Software Distribution on Your Workstation . . . . .	178
stat – Check the Status of a Communication Queue . . . . .	179
stattg – List Target Status . . . . .	182



	BACKUP PROGRAM Keyword . . . . .	273
	REMOVE PROGRAM Keyword . . . . .	274
	ACCEPT PROGRAM Keyword . . . . .	274
	UNINSTALL PROGRAM Keyword . . . . .	274
	MAINTENANCE SYSTEM Keyword . . . . .	275
	Structure of a CID Software Profile . . . . .	275
	Example of a CID Software Profile . . . . .	278
	<b>Index</b> . . . . .	<b>279</b>

---

## Figures

1.	The structure of a Change File profile . . . . .	250
2.	Example of a TME 10 Software Distribution Change File profile . . . . .	267
3.	Example of a TME 10 Software Distribution Change File profile . . . . .	268



## **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 . . . . .	10
4.	Authorization Levels for Command Use . . . . .	16
5.	Default File System IDs . . . . .	49
6.	Object Types and their SNA/FS Class Codes . . . . .	49
7.	Object Types and their SNA/FS Class Codes . . . . .	107
8.	Default File System IDs . . . . .	174
9.	Values for Component Names . . . . .	188
10.	Updating User Authorizations to Targets . . . . .	228
11.	Command Attributes Required for Each Command Name . . . . .	246



---

## 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:

<b>Server Scratch</b>	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.
<b>Server Upgrade</b>	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.
<b>Client Scratch</b>	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.

*Table 1. TME 10 Software Distribution, Version 3.1.5 Platform Support*

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
<b>TME 10 Software Distribution</b>		
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
<b>NetView DM/2</b>		
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 MVW 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
- How to create a change file for installing non-CID applications using DiskCamera commands. (see 233)
- A summary of how user-group authorization affects the availability of commands

## Help Command

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

```
nvdms 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:

```
nvdms help rentg
```

## Configuration Commands

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

**addgp (see 29)**

Define a group of targets

**addpm (see 31)**

Add parameters to a target definition

**addprf (see 33)**

Create a new authorization profile

**addtg (see 36)**

Add a new target workstation to the system

**addusr (see 41)**

Create a new TME 10 Software Distribution user

**delgp (see 63)**

Delete a group of targets

**delpm (see 65)**

Delete parameters from a target definition

**delprf (see 66)**

Erase a user authorization profile

**deltg (see 69)**

Remove a target definition

**delusr (see 70)**

Delete a user definition

**inv (see 99)**

Inventory discovery

**lsak (see 109)**

List names and descriptions of data or target access keys

**lsbs (see 110)**

List the server configuration

**lsgp (see 117)**

List groups of targets

**lsprf (see 119)**

List the contents of a user authorization profile

**lstg (see 129)**

List target configurations

**lsusr (see 133)**

List user authorization levels

**rentg (see 153)**

Rename a target workstation

**updak (see 205)**

Update a data or target access key

**updcn (see 207)**

Update the change management status for an object at a target

**updbn (see 206)**

Change the server configuration

**updprf (see 212)**

Update a user authorization profile

**updpwd (see 215)**

Update a user's password

**updtg (see 220)**

Update the target configuration

**updusr (see 227)**

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 47)**

Build a change file

**cat (see 48)**

Add an entry to the catalog

**del (see 54)**

Delete a file

**delf (see 59)**

Delete a noncataloged file

**exp (see 82)**

Export a change file

**imp (see 91)**

Import and catalog a change file

**ls (see 107)**

List entries in the catalog

**lscf (see 111)**

List the contents of a change file

**rtrv (see 158)**

Retrieve a file from a target

**rtrvf (see 163)**

Retrieve a noncataloged file from a target

**send (see 168)**

Send a file to a target

**sendf (see 173)**

Send a noncataloged file to a target

**unbld (see 195)**

Unpack the components of a change file

**uncat (see 198)**

Remove an entry from the catalog

**upd (see 204)**

Update a catalog entry

**updcmm (see 207)**

Update the change management status for an object at a target

## Change Control Commands

Use these commands for change control requests.

**acc (see 20)**

Accept a previously installed change file on a target

**act (see 25)**

Activate changes on a target

**auth (see 43)**

Authorize a change file for use by a target

**delcm (see 58)**

Clear the status of change files

**exec (see 74)**

Execute a command procedure on a specified target

**execf (see 78)**

Execute a noncataloged program or script on a target

**inst (see 92)**

Request the installation of a change file on a target

**lscm (see 114)**

List the status of change files at one or more targets

**rem (see 148)**

Remove a previously installed change file from a target

**unauth (see 191)**

Remove the authorization for a target to use a change file

**uninst (see 199)**

Completely remove a component from a target

**vercm (see 229)**

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 90)**

Hold a push-mode target

**log (see 104)**

View the message log

**reset (see 155)**

Reset the product

**rld (see 156)**

Prompt all components to reread the configuration

**reltg (see 147)**

Release a push-mode target

**start (see 178)**

Start the product

**stat (see 179)**

Report status of TME 10 Software Distribution communication

**stattg (see 182)**

List the status of targets

**stop (see 185)**

Stop the product

**svr (see 186)**

Select which server to operate on

**trace (see 187)**

Set internal tracing

**troff (see 189)**

Stop tracing

**tron (see 190)**

Start tracing

## Request Management Commands

Use these commands to manage requests.

**delrq (see 67)**

Delete a submitted request

**eraserq (see 72)**

Erase a request from the database

**hldrq (see 88)**

Hold a request

**lsrq (see 124)**

List all requests on the database

**reirq (see 145)**

Release a request

**rstrq (see 157)**

Restart a request

**updrq (see 216)**

Update a request

## Command-Line Interface

### Queue Management Commands

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

**hldc (see 85)**

Hold remote communication

**hldq (see 86)**

Hold processing distributions on a queue

**lsq (see 121)**

List the distributions waiting on a queue

**prgq (see 137)**

Remove a distribution from a queue

**prtyq (see 139)**

Increase the priority of an entry in a queue

**relc (see 142)**

Release remote communication

**relq (see 143)**

Release processing distributions on a queue

---

### Entering a User ID and Password

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

- Enter your user ID as environment variables using the FNDUSER keyword. At a command line, enter:

```
set FNDUSER=<userid>
```

where <userid> is your user ID.

You can set the value of FNDUSER to \* (asterisk), meaning that every time you enter a command you are prompted for your user name. If you do not set any value for FNDUSER, it defaults to the user ID you used to log on to the operating system.

- Enter your user ID as parameters when you enter a command:

```
nvdm cmd_name <cmd_parameters> -u<user_name>
```

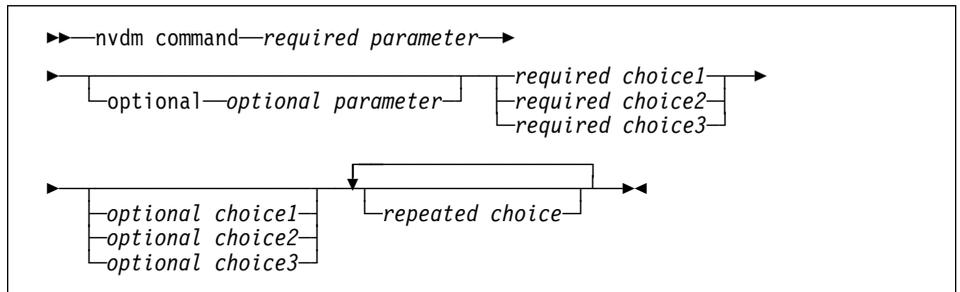
- Open an *interactive command line session* by entering the **nvdm** command. When you issue this command, you are prompted to specify your user ID. You then no longer need to supply this information for each command you enter. The interactive command line session stays open until you close it by issuing the **quit** command. See “nvdm – Start a Command Line Session” on page 135 and “quit – Close an Interactive Command Line Session” on page 141.

---

### 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.

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



**nvdms** 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.

## Command-Line Interface

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.

<b>Symbol</b>	<b>Description</b>	<b>Symbol</b>	<b>Description</b>
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		

## 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 12). 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.

## Command-Line Interface

### \$(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**)

**-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.

## Command-Line Interface

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:

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

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.

```
nvdminst 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).

## 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.
- 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

## Command-Line Interface

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.

### Authorization Levels for Command Use

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

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

*Table 4 (Page 2 of 2). Authorization Levels for Command Use*

<b>Authorization Level</b>	<b>Commands Allowed</b>
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 uppbs, lsbs updusr, lsusr
Send - Yes	ls, send, cat
Retrieve - Yes	ls, rtrv
Queues - View	lsq stat
Queues - Modify	hldc, relc lsq prgq, hldq, relq stat
Change Management Install - Yes	inst lscf lsrq, lscm
Change Management Execute - Yes	exec ls lscm
Change Management Authorize - Yes	auth, unauth ls lscm, delcm

## Command-Line Interface

---

## Chapter 2. TME 10 Software Distribution Line Commands

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

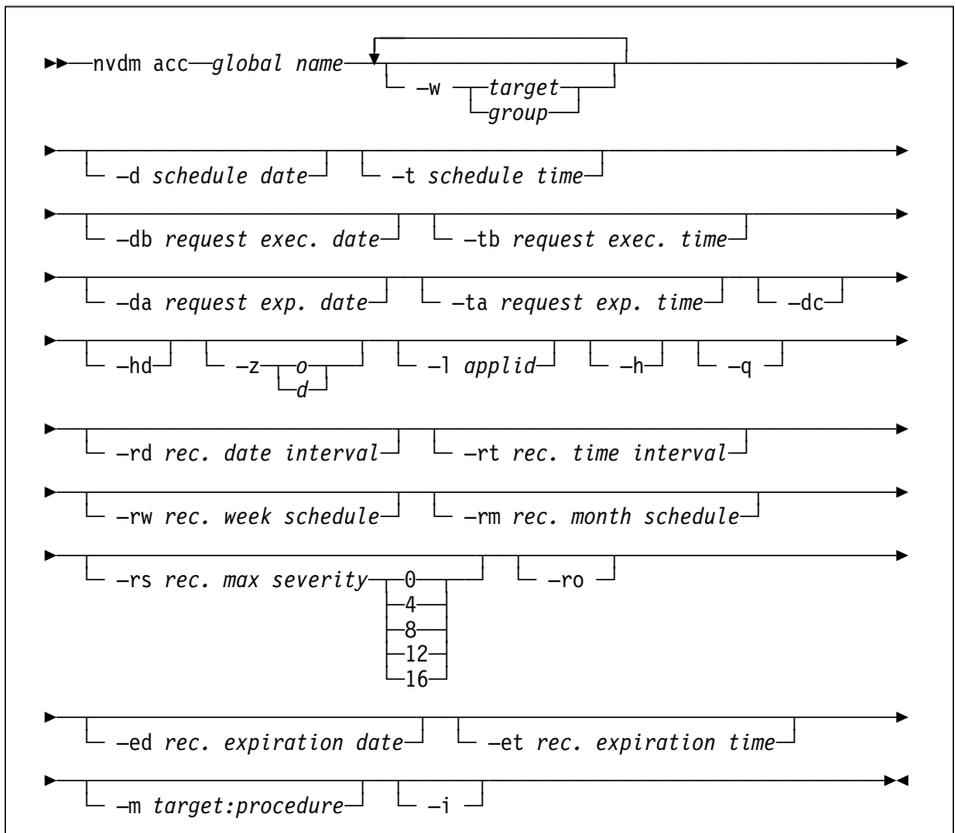
## acc – Accept an Installed Change File

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 12.

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 11.

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 12.

**-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 15.

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**

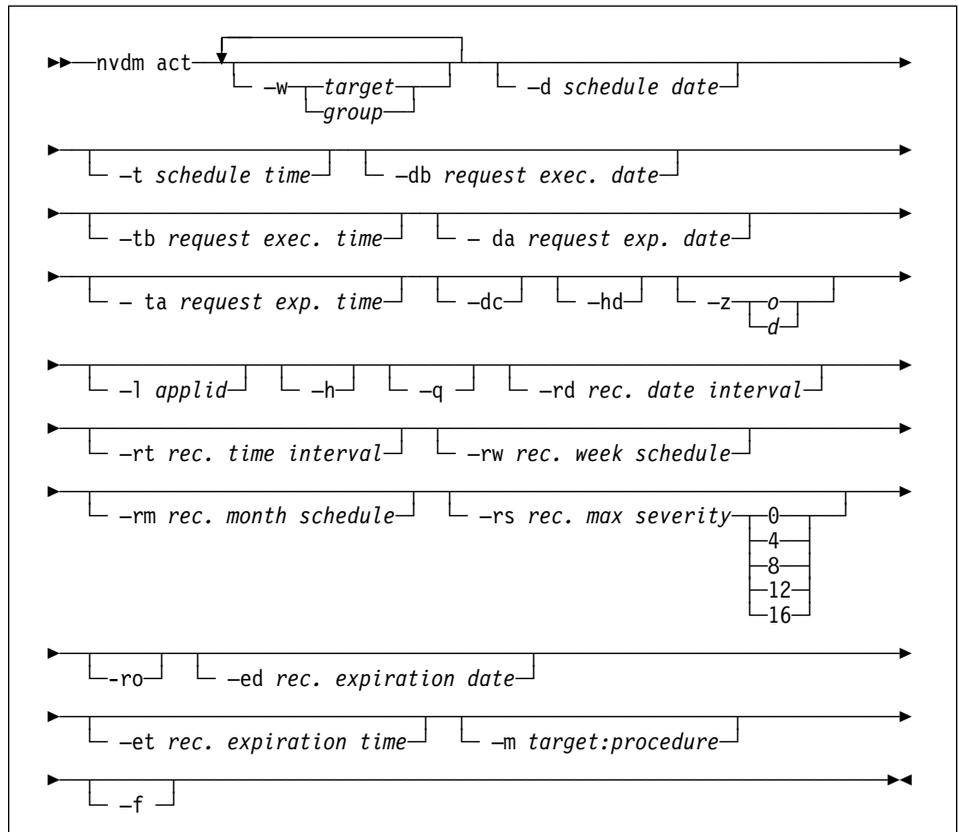
The lsrq, lscm, inst, rem, and uninst commands.

## 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 11.

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 12.

**-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 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 15.

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 FREDSWS -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 FREDSWS is not quiesced at the time the activation is to be performed, the activation will not take place.

## Related Information

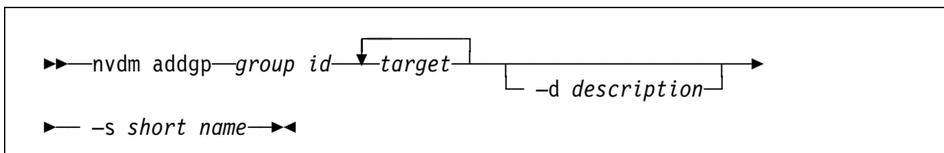
The `inst`, `rem`, `uninst`, and `lsrq` commands.

## 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



### 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 FREDSWJ JOHNSWS -d "Word Processor Users"
```

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

## **addgp**

### **Related Information**

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

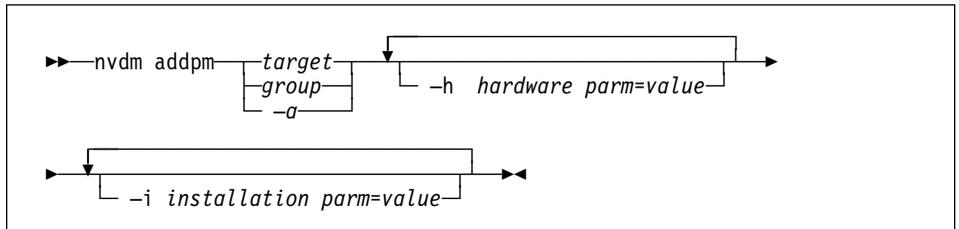
## addpm – Add Parameters to a Target

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.

## addpm

### **-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

```
nvdn addpm FREDWS -i dir1=c:\user\mydir
```

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

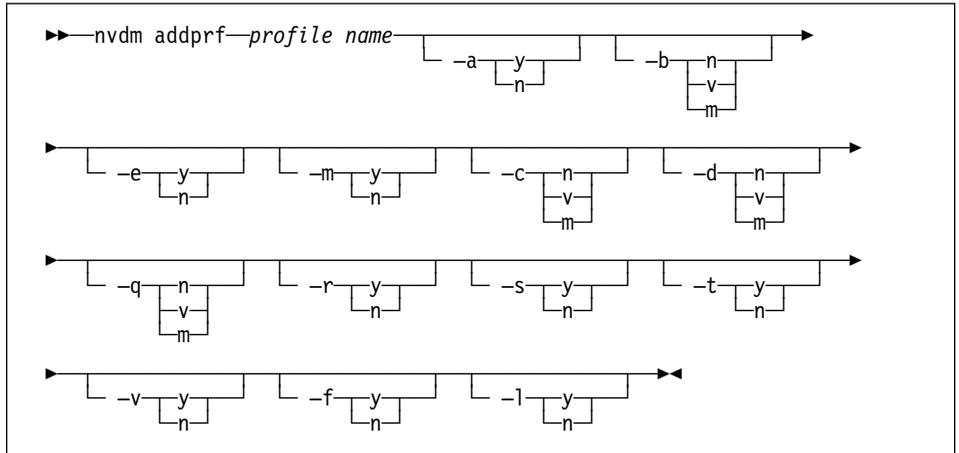
## Related Information

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.

## addprf

**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.

**-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 EUROSALES -an -bn -en -my -cv -dn -qv -ry -sy -tn -vy -fn -ln
```

This command creates a user profile, EUROSALES, 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

EUROSALES users are *not* authorized to:

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

EUROSALES users can view:

- Network configuration information
- Queues

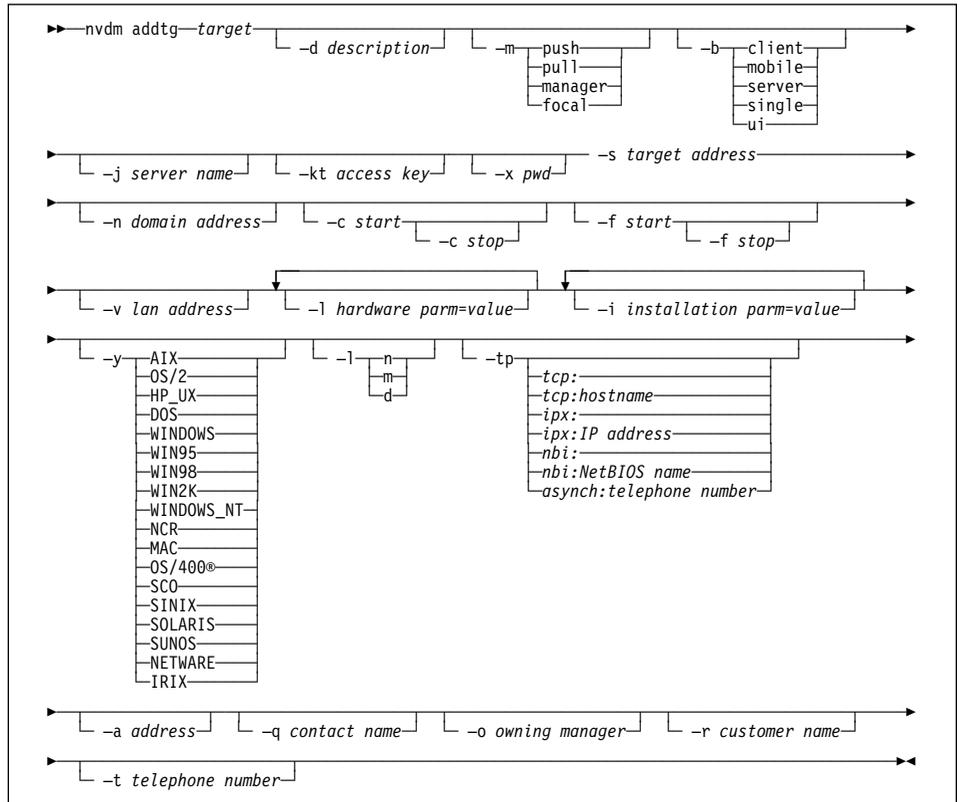
**Related Information**

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.

**ipx:** The transmission protocol is ipx.

**ipx:IP address**

The transmission protocol is IPX, with the IP address you specify.

**nbi:** The transmission protocol is NetBIOS.

**nbi:IP address**

The transmission protocol is NetBIOS, with the NetBIOS name 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.

## addtg

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.

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 FREDSWs -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 WINDOWS NT -tp tcp
```

This command defines the remote target FREDSWs. 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 WINDOWS NT operating system. It communicates with the server across a TCP/IP link.

## Related Information

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

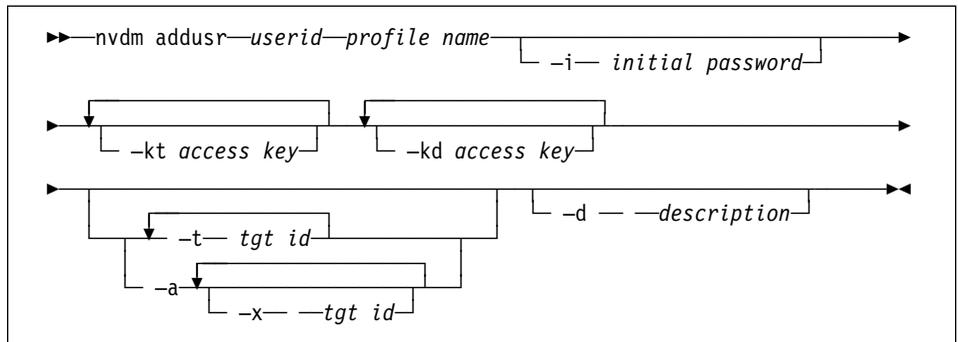
## addusr – Create a User

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.

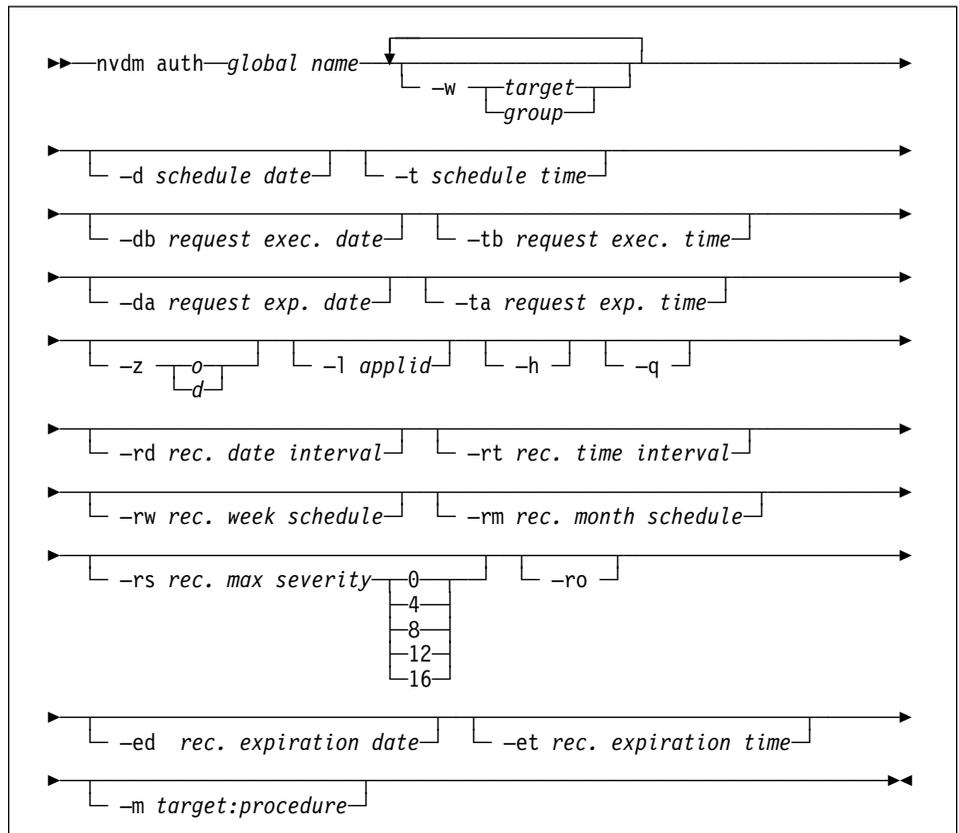
## auth – Authorize Targets to Access Files

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 12.

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 11.
- 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 12.
- 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 15.

## 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 249). 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—┘
```

### 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.

### Examples

```
nvdm bld c:\user\mybuild\profile
```

This command builds, catalogs, and sends to the server the change file specified in the change file profile named `c:\user\mybuild\profile`.

### Related Information

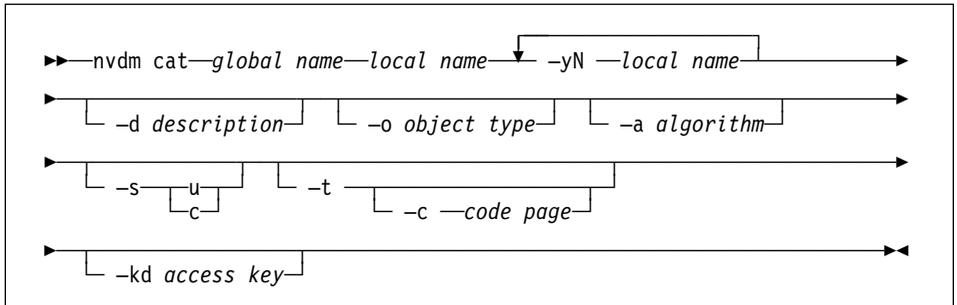
The `cat`, `lscf`, and `unbld` commands. Chapter 3, “Creating Change File Profiles” on page 249.

## 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.

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 c:\user\amc\s 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 c:\user\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 megafast compression algorithm with the **-z** parameter.

**Related Information**

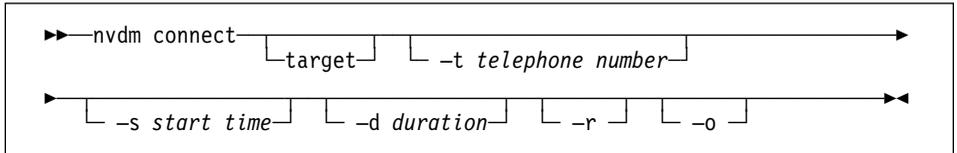
The uncat command.

---

## 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 nvdms.cfg file. See the *TME 10 Software Distribution for OS/2 Quick Beginnings* for more details on the nvdms.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

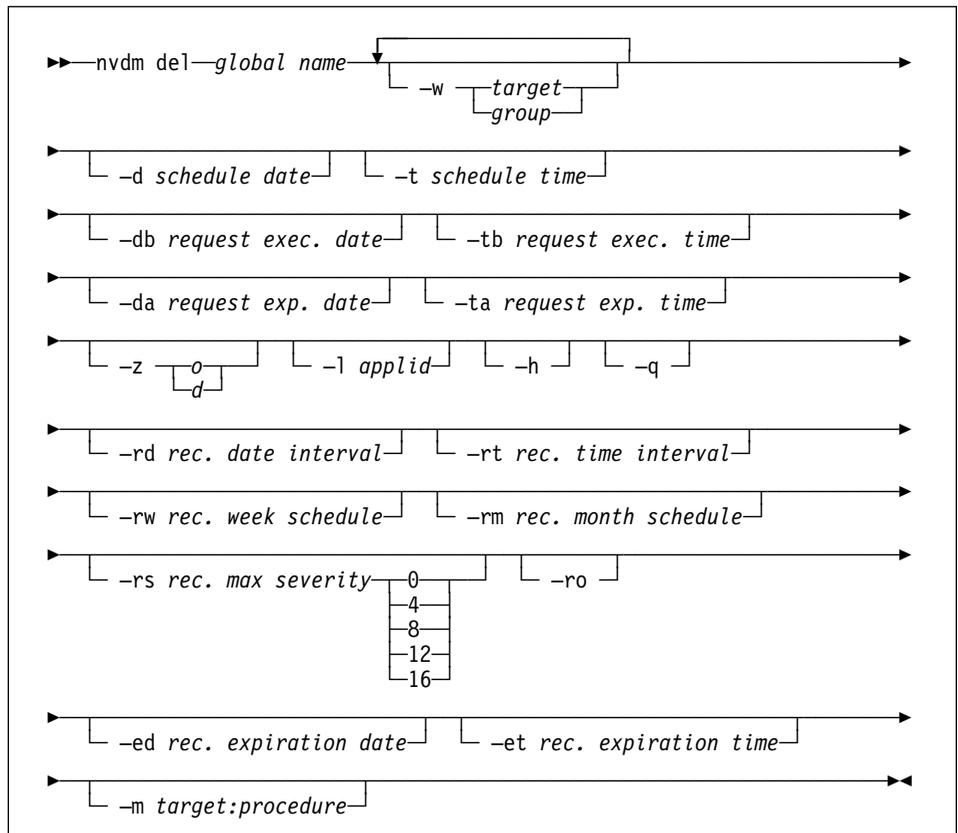
Use the **del** command to erase a file from a local or remote target or group. The file is identified by its global name. You can initiate delete operations on remote targets or groups only from a remote administration site.

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 12.

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 11.
- 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 12.
- 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	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:

<b>0</b>	Successful
<b>4</b>	Warning
<b>8</b>	Error
<b>12</b>	Failed
<b>16</b>	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 15.

## Examples

```
nvdms del MYBUILD.FIX -w FREDSWS
```

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



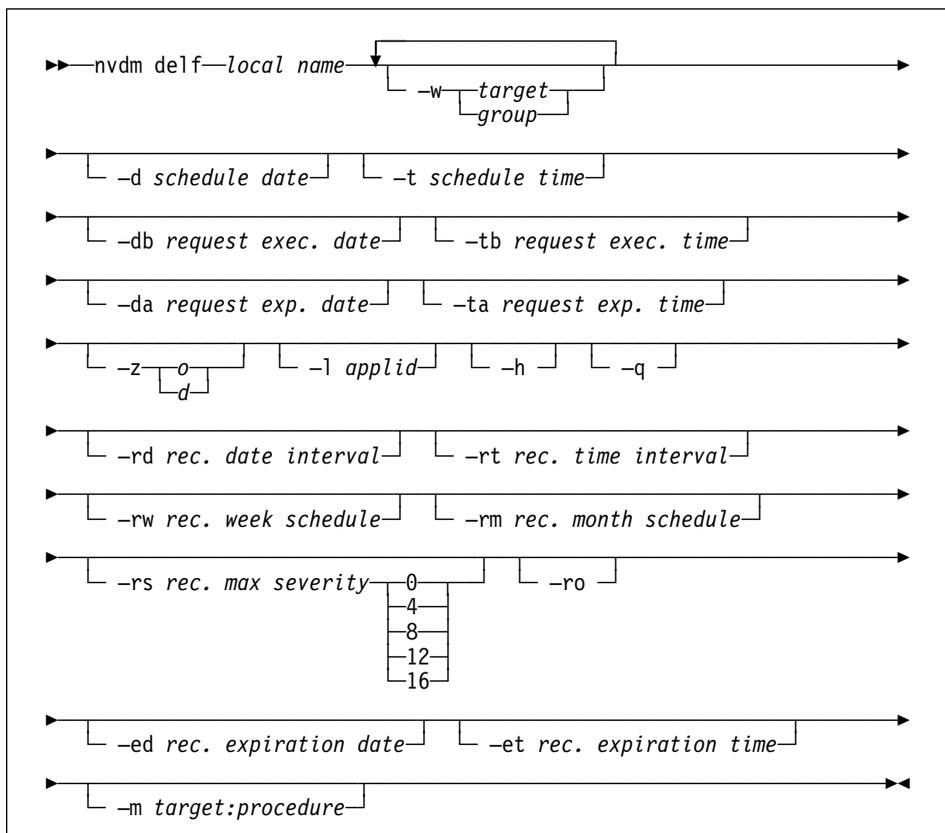
## delf – Delete a Noncataloged File

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 11.

**-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 12.

**-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 15.

## 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.

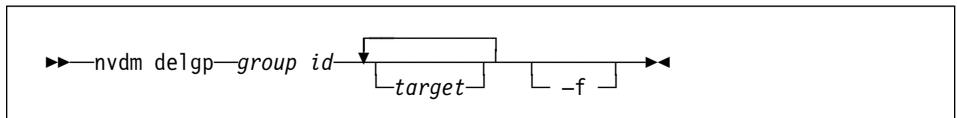
## delgp – Delete a Target Group

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



### 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 FREDSSWS HARRYSWS
```

This command removes the targets `FREDSSWS` and `HARRYSWS` from the group `WORD`. No confirmation is required.

**delgp**

**Related Information**

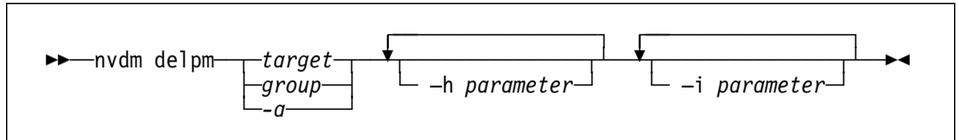
The addgp and lsgp commands.

## 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

---

### 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

**-w** *target | group*

The targets or groups where the request is to be deleted. See “Entering Target Names” on page 11.

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

```
nvdms 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 FREDSSWS
```

This command deletes the target FREDSSWS from the network.

## Related Information

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

## delusr

---

### 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 `updusr` 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

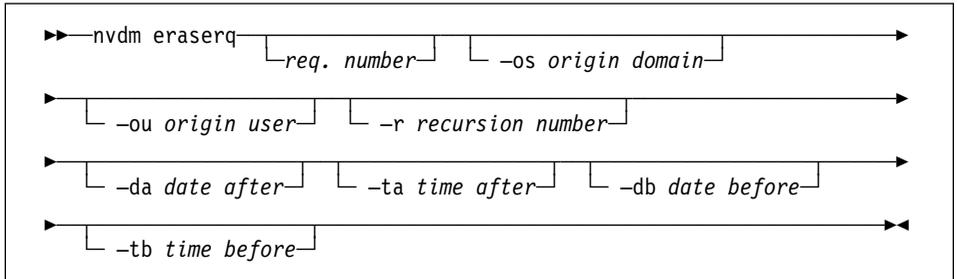
The name of the target you want to disconnect from the server.

- 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

```
nvdms 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>
```

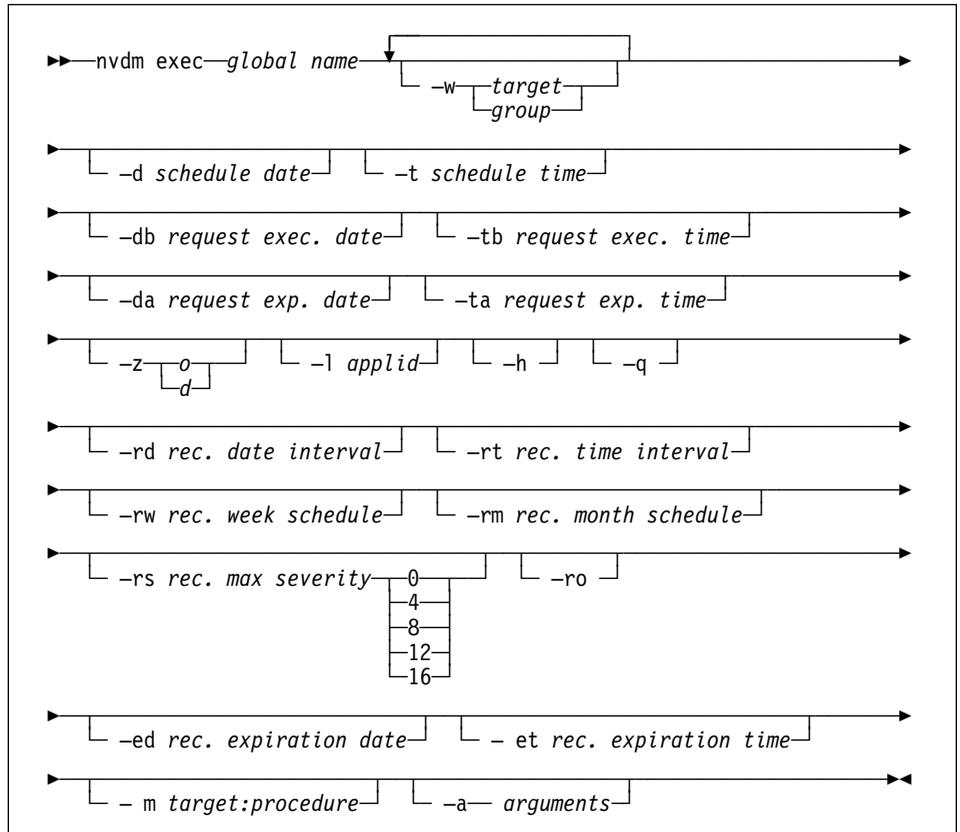
to ensure that the correct connection is made before commands are executed.

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 12. 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 11.

### **-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 12.

### **-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 15.
- 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 FREDSWS -a "c:\user\bin"
```

This command executes the program or script known by the global name EURO.VIRUSCHK.EXE.1.US on FREDSWS, using the parameter C:\user\bin. The procedure is executed as soon as possible.

## Related Information

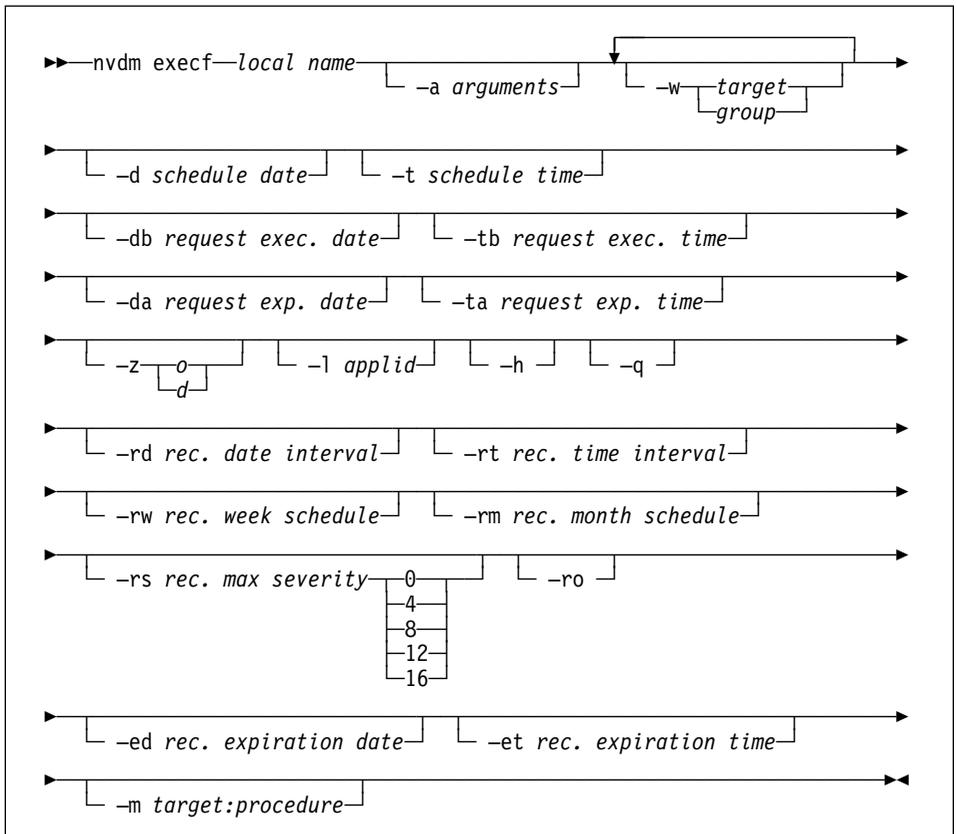
The lsrq command.

## 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 11.
- 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 12.
- 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 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 15.

## Examples

```
nvdms 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.

```
nvdms 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,

```
nvdms 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.

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

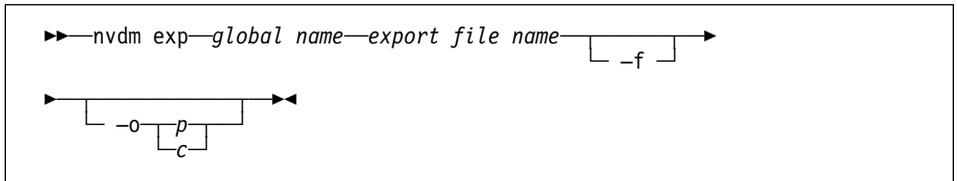
This command causes the program PGM1.EXE to be executed at target2 and target3, which are connected to an NT server.

## 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 12.

#### *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 c:\u\me\wordproc
```

This command copies the change file EURO.WORDPROC.REF.1.US to file c:\u\me\wordproc on your computer. The file c:\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

---

### 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.

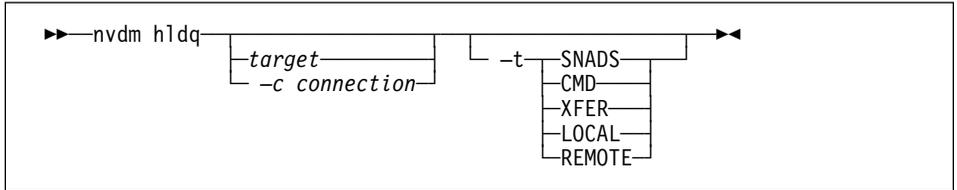
### Related Information

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 C:\softdist\db\snacon 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 `hlcd` commands.

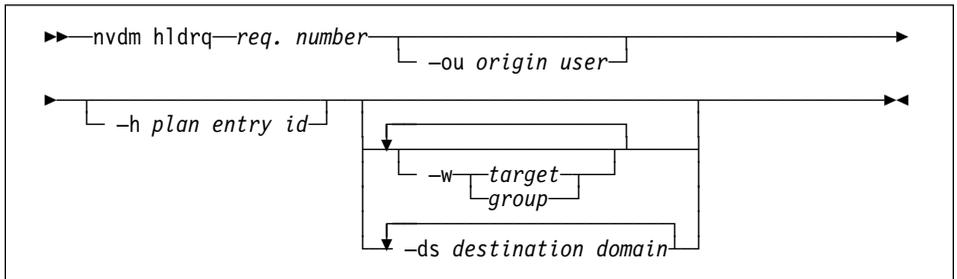
## hldrq – Hold a Request

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 11.

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**

```
nvdn hldrq 13 -w FREDSSWS
```

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

**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 FREDSWS
```

This command holds the target named FREDSWS.

### Related Information

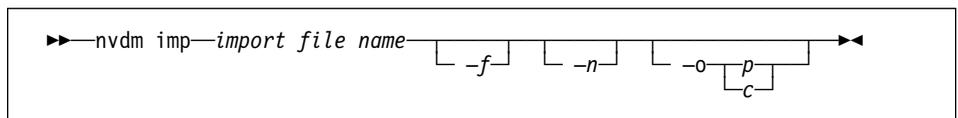
The `reltg` command.

## imp – Import a Change File into the System

Use the **imp** command to import a noncataloged change file 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



### 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 c:\u\me\amword
```

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

### 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:

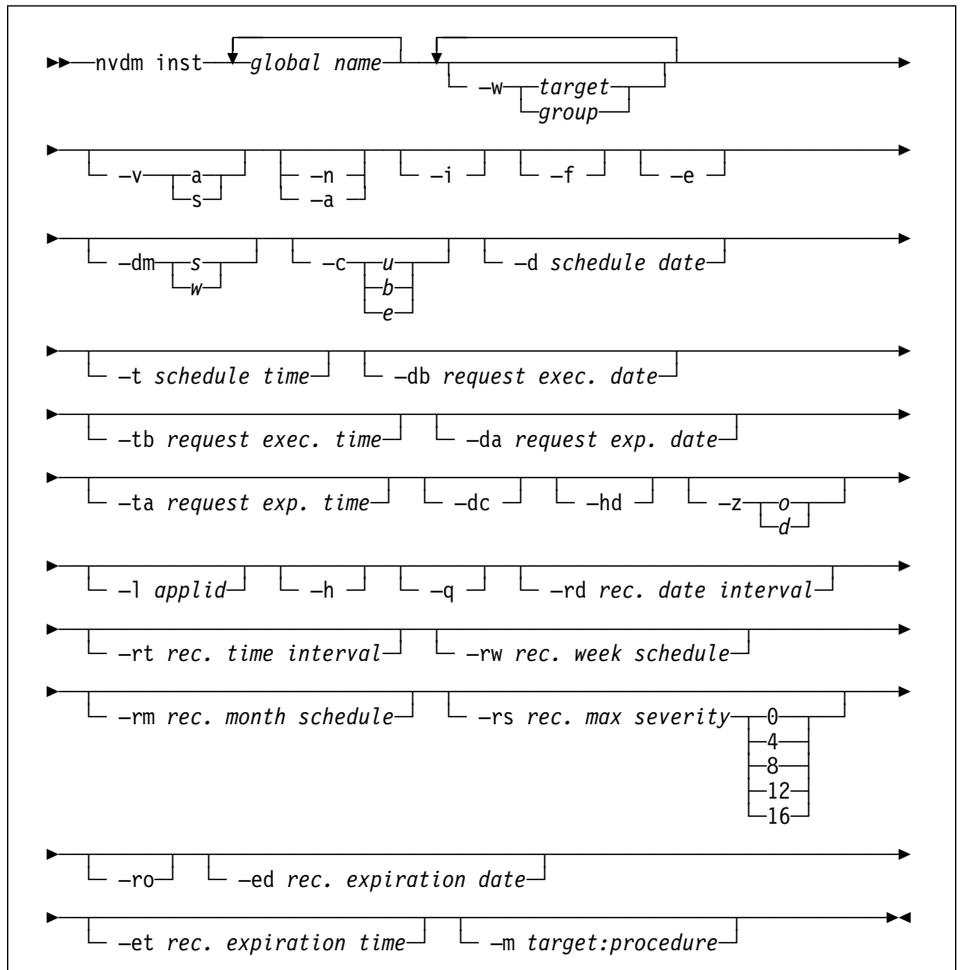
- The target or group on which the installation is to be performed.
- Whether the installation will be completed by an activation, in which case the installation is done to the service area. If an activation is not required, the installation should be done direct to the active area.
- Whether the change 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 11.

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 `fndlog` 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 12.

**-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:

<b>0</b>	Successful
<b>4</b>	Warning
<b>8</b>	Error
<b>12</b>	Failed
<b>16</b>	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 15.

This parameter cannot be used locally by mobile clients.

## inst

### 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 `fndswinv`, `fndhwinv`, and `fndtkinv` files at the target where the inventory is performed.

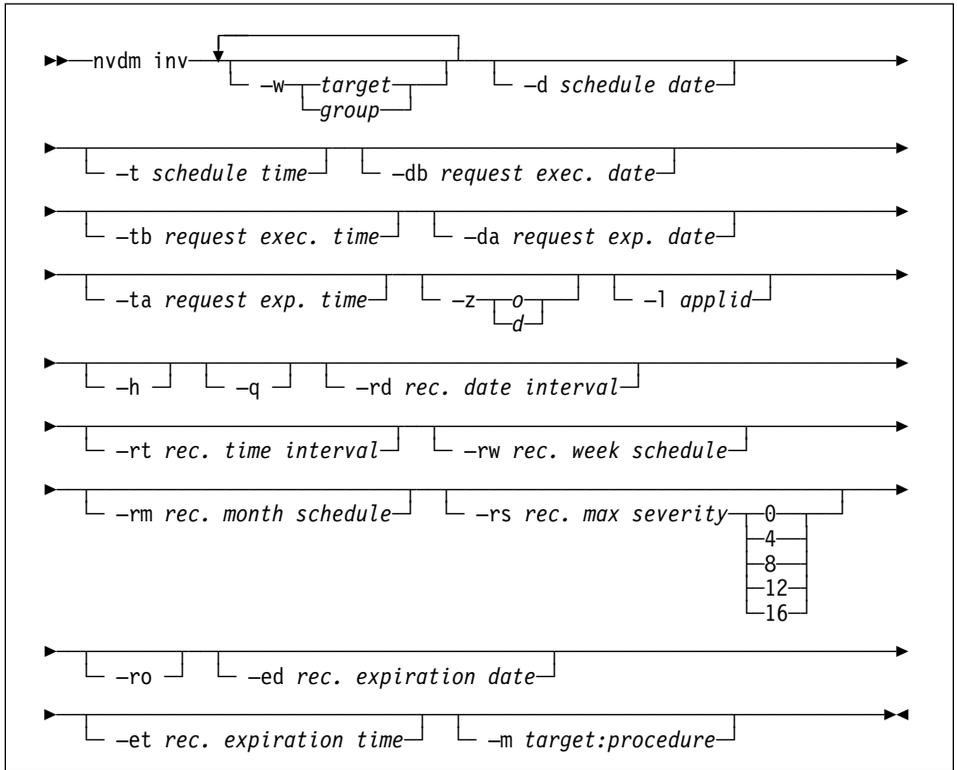
This operation can be performed on remote targets only if the targets involved in the operation are connected by the server-to-server (STS) transmission protocol. This procedure is normally executed by a script after running the inventory discovery program. You need Modify Configuration authorization 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 11.

**-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 12.

**-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 15.

## Examples

```
nvdms 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

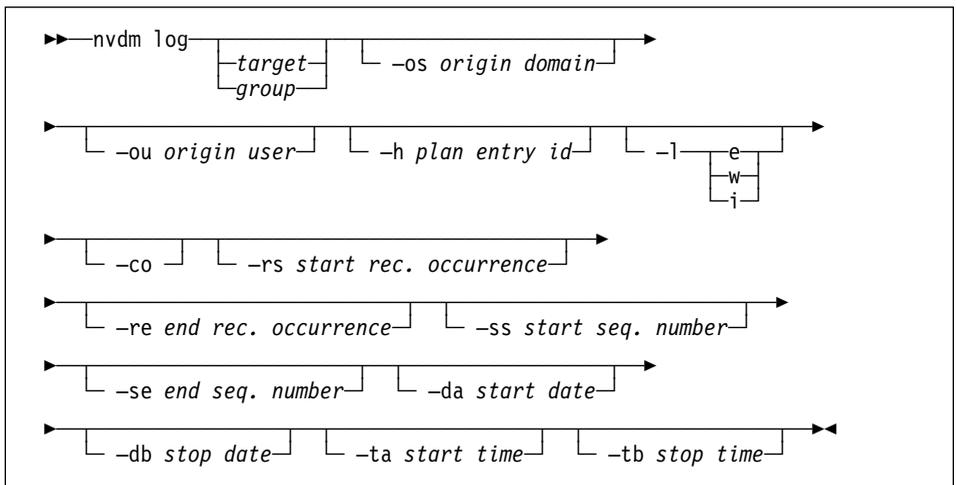
Use the **log** command to display messages contained in the message log. You can use parameters to limit the messages displayed to ones that meet certain criteria.

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.

## log

**-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 12.

**-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

```
nvdn log -w FREDSWS -1w
```

This command shows all messages issued from FREDSWS 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

```
nvdm ls—global name [ -l ] [ -o object type ]
```

### Parameters

*global name*

The global name of the entries to be listed. See “Entering Global Names” on page 12. 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.

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

## Is

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: c:\user\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: c:\user\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)
```

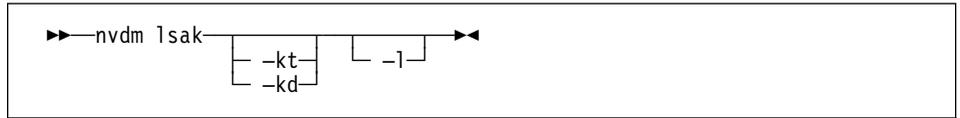
## Related Information

The cat, imp, and bld commands.

## Isak – List Access Keys

Use the **Isak** 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

```
nvdmsak -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

---

### 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 updbs command.

## Isctf – List the Contents of a Change File

Use the **Isctf** 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

```

▶─nvdm Isctf─┬─global name─┬─▶
              └─local file name─ -n ─┬─ -1 ─┬─▶

```

### 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

# lscf

## 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:        C:\softdist\fndnoss
TARGET NAME:        C:\softdist\fndnoss
TYPE:               FILE
ACTION:             COPY
INCLUDE SUBDIRS:    NO
OWNER:              root
GROUP:              FNDADMIN
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.

**Related Information**

The `bld` command.

Chapter 3, “Creating Change File Profiles” on page 249.

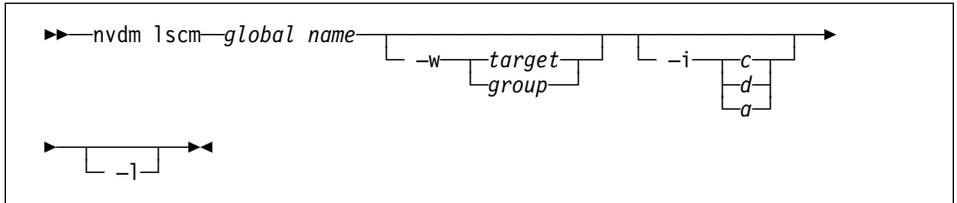
---

## 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 12. 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 11.

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 **-l** if **-i** is specified because the information is related to change files only.

## Output

The output of the **Iscom** 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

```
nvdm 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:      FREDSW
                  Status:      Available
                  Target:      JOHNWS
                  Status:      Installed, removable, active

Global name:      EURO.WORDPROC.REF.2.US
                  Target:      FREDSW
                  Status:      Installed, removable, active
```

## Related Information

The delcm command.



## lsgp

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

                FREDSW          Fred's Workstation
                JOHNSWS         John's Workstation
                BACKUP           Backup Workstation
```

## Related Information

The addgp, addpm, delgp, and delpm commands.



# lsprf

```
nvdm 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.

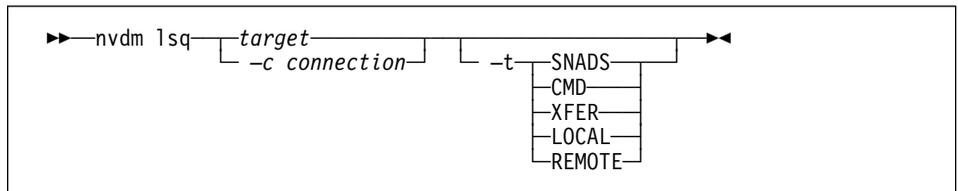
## Isq – Display Queue Entry Details

Use the **Isq** 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 C:\softdist\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: FREDSWS  
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

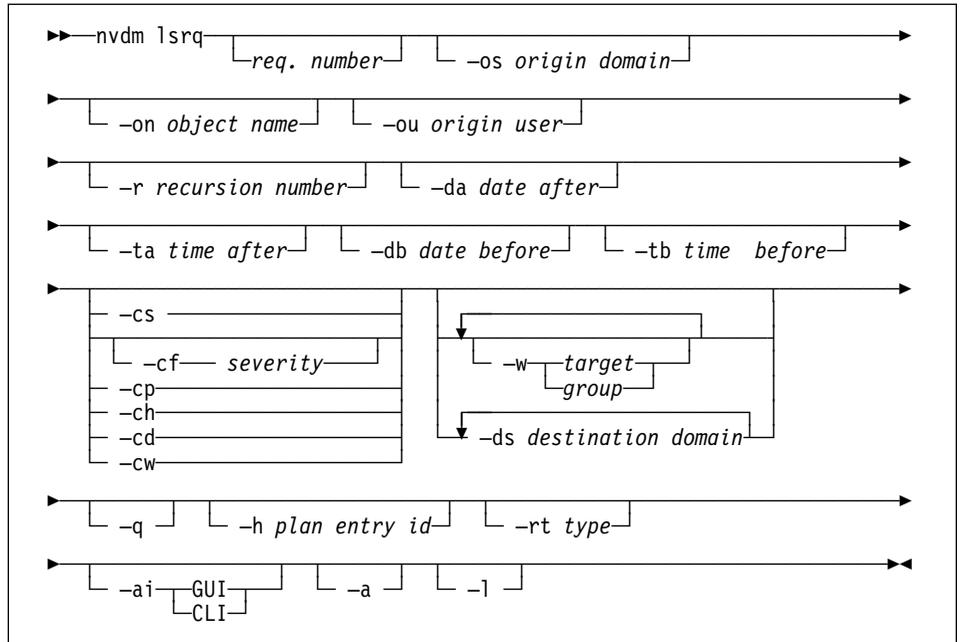
The stat, hldq, relq, and prgq commands.

## 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 11.

**-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:

```
nvdn 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
```

## Isrq

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
```



**-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 **istg** 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:          FREDSWS
Mode:           Pull
Description:    Fred's Computer
Type:          CLIENT
```

```
nvdms istg '*' -l
```

This command lists all targets in long mode. The output looks like the following.

```
Target:          FREDSWS
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=c:\user\apps\nvd
                   word=c:\user\apps\word

Hardware parms: (none)
Discovered Inventory: (none)
```

## Related Information

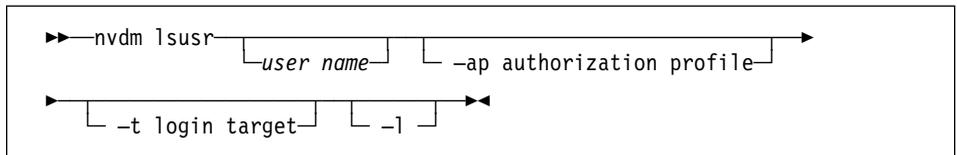
The addtg, deltg, updtg, stattg, and lsgp commands.

## Isusr – List User Authorizations

Use the **Isusr** 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 Isusr 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

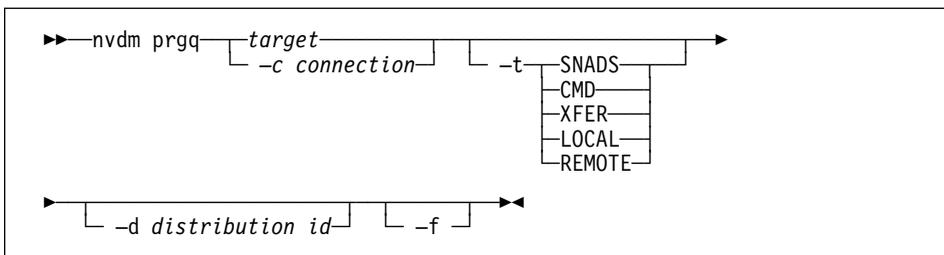
The `nvd`, `svr`, and `quit` commands.

## 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 C:\softdist\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

## prgq

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

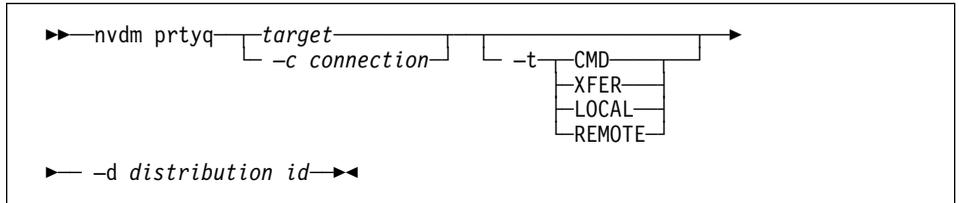
The `lsq`, `hldq`, and `relq` commands.

## prtyq – Give Priority to a Queue

Use the **prtyq** 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 C:\softdist\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.

## **prtyq**

### **Examples**

```
prtyq FREDSW -d 3
```

This command gives priority to distribution 3 for target FREDSW.

### **Related Information**

The lsq command.

---

## 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 `nvdms` and the `open` command.

## relc

---

### relc – Release Remote Communication

Use the **relc** command to release all remote communication, undoing the effect of the **hlcdc** 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

```
nvdms relc
```

This command releases the SNA network communication to and from the server.

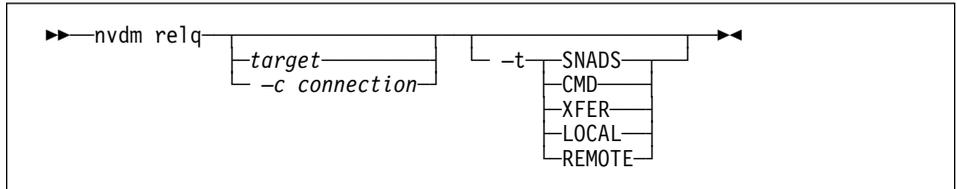
### Related Information

The `hlcdc` 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, under the `C:\softdist\db\snacon` 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.

**relq**

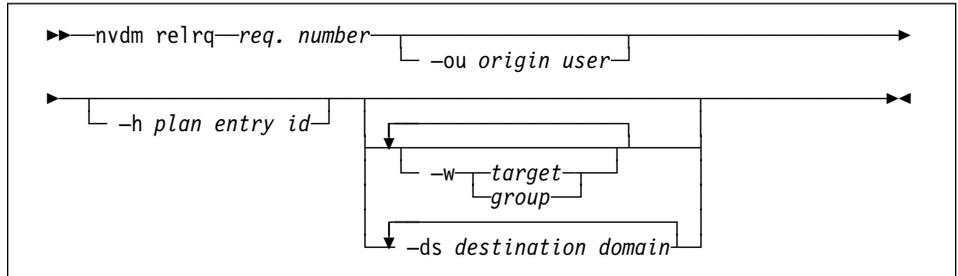
### **Related Information**

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 11.

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.

**relrq**

### **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

```
nvdm reltg FREDSSWS
```

This command releases the target named FREDSSWS.

### Related Information

The `hldtg` and `stattg` commands.

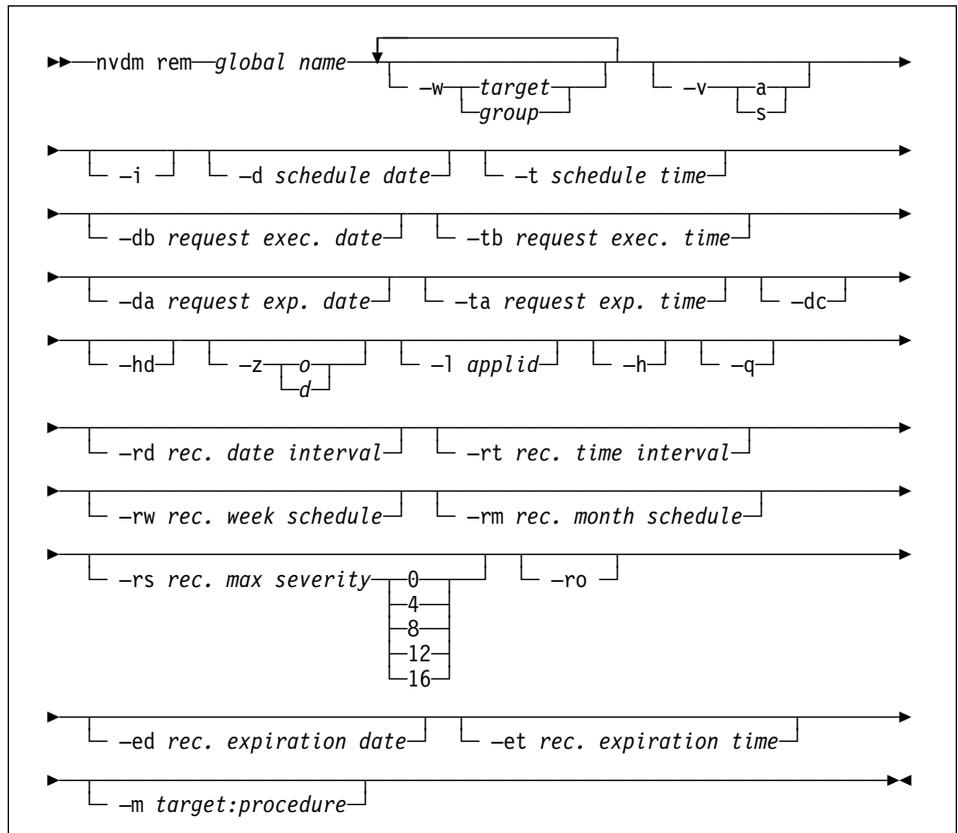
## rem – Remove a Change File From a Target or Group

Use the **rem** command to request that a previously installed change file be removed from a target or group, thereby restoring the component to the state it was in before the installation was performed. In order for the **rem** command to be accepted, the following conditions must be met:

- The catalog entry for the change file must still exist (although the actual file may have been deleted).
- The change control status must be one of the following:
  - Installed, removable, active
  - Installed, removable, inactive
  - Installed, removable, reboot required

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 remove. See “Entering Global Names” on page 12.

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 11.

This parameter cannot be used locally by mobile clients.

### **-v** *ajs*

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 12.

### **-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 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.

## rem

**-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 15.

This parameter cannot be used locally by mobile clients.

## Examples

```
nvdn rem EURO.WORDPROC.REF.2.US -w FREDSWS
```

This command removes the previously installed change file `EURO.WORDPROC.REF.2.US` from target `FREDSWS`, 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 `C:\softdist\nvdm.cfg` file to reflect the server's new name
- Erase all of the configuration files located in the `C:\softdist\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.

**rentg**

**Related Information**

The addtg, deltg, lstg, and updtg commands.

---

## 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
```

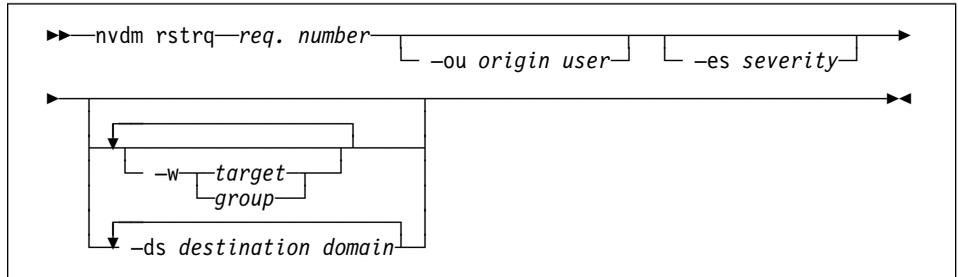
Causes remote connection configuration files to reload their configuration data.

## 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 11. 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 FREDSSWS
```

This command restarts the request whose number is 13 for target FREDSSWS.

### 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
```

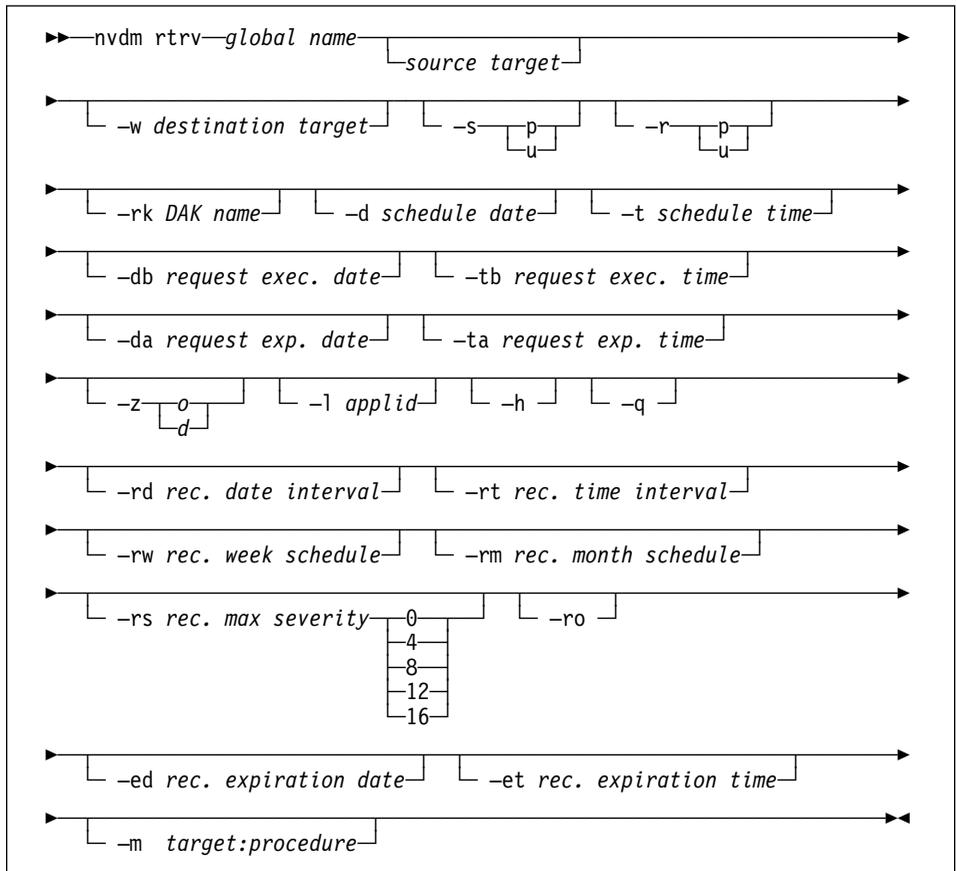
where 0 and 1 have the following meaning:

- 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 12.

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 12.

**-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
```

## rtrv

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 15.

## Examples

```
nvdm 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.

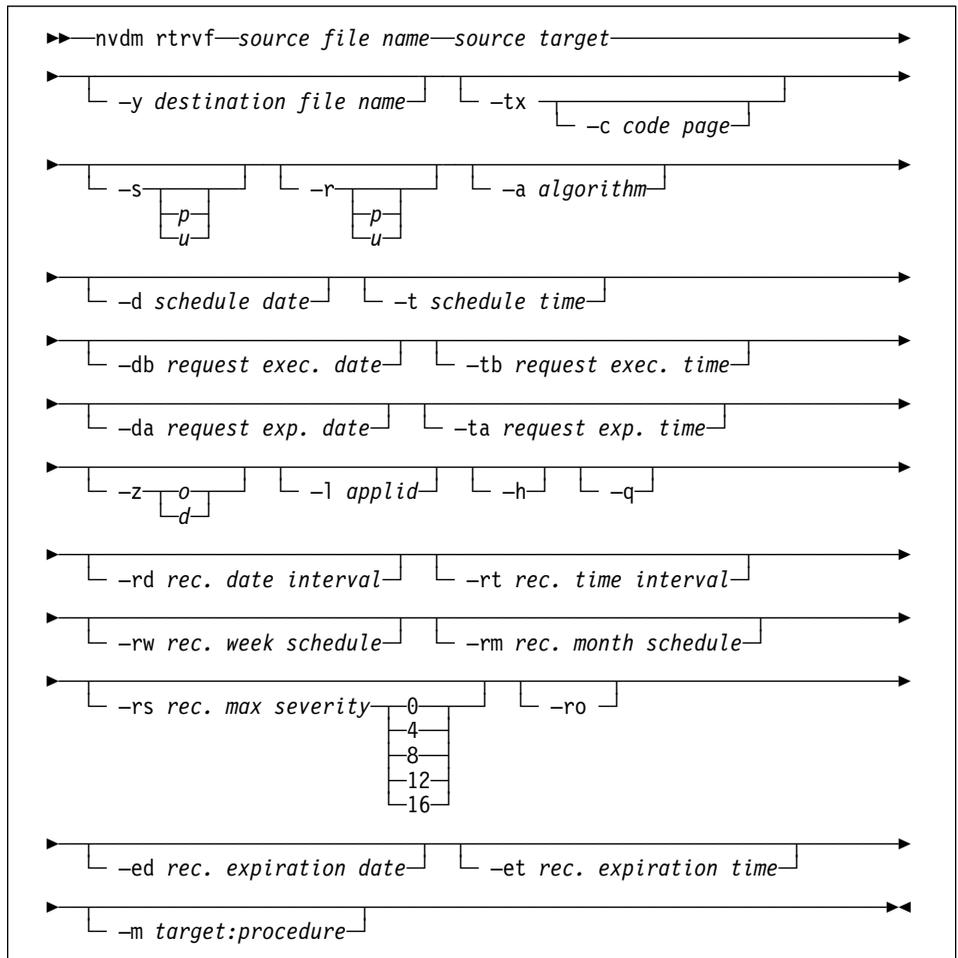
## rtrvf – Retrieve a Noncataloged File

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 `c:\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 `C:\softdist`.

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 `C:\softdist\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 12.

**-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.

**-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 15.

## Examples

```
nvdms 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:

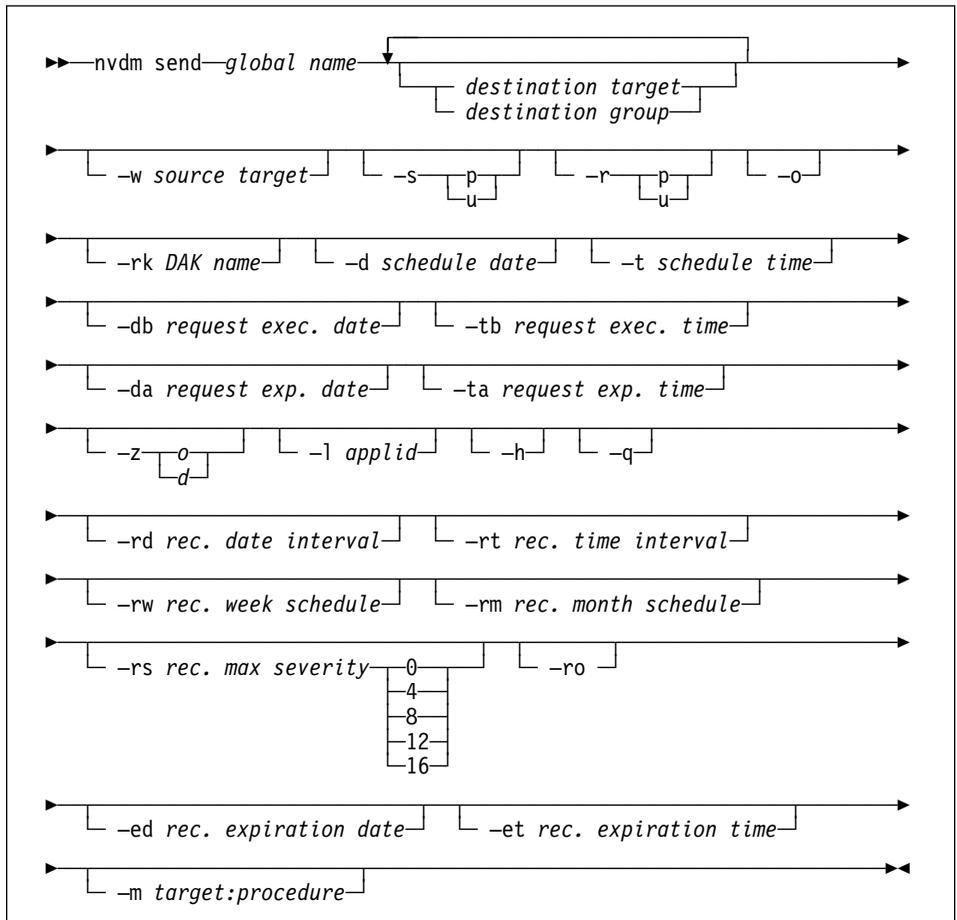
- 0** When a remote server sends a file in compress format and wants to store it in decompress format at destination target, the file is decompressed at local server and sent in decompress format to destination target.  
  
When a remote server retrieves a file in compress format the file is compressed 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.

## Syntax



## Parameters

### *global name*

The global name of the file. See “Entering Global Names” on page 12.

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 11. If no **-w** parameter is specified, the file is assumed to be stored on your computer.

## send

### **-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 12.

### **-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.

## send

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 15.

## Examples

```
nvdmsend EURO.SPELLCHECK.EXE.1.US FREDSW
```

This command causes the file EURO.SPELLCHECK.EXE.1.US to be sent to the target FREDSW. 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.

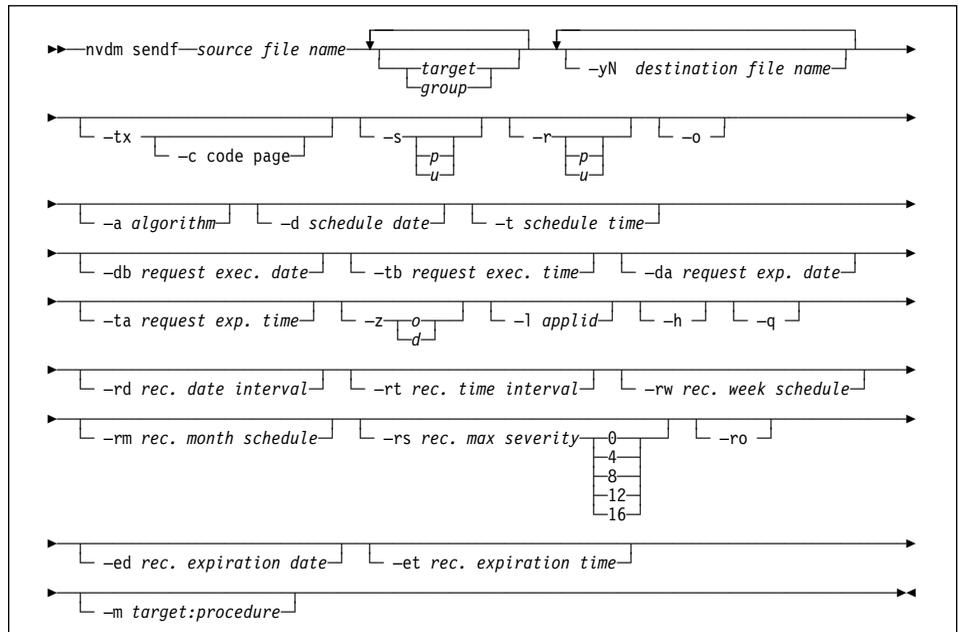
## sendf – Send a Noncataloged File

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 p|u**

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 12.

**-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

## sendf

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 15.

## 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 – 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.

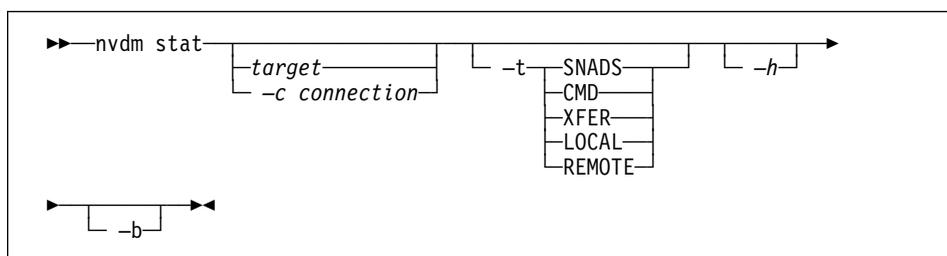
## stat – Check the Status of a Communication Queue

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.

# stat

## **-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 C:\softdist\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

```
nvdn stat target1
```

The output of this command looks like the following:

```
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.

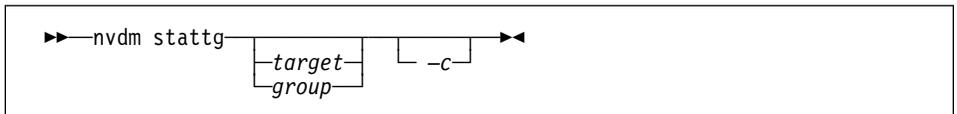
## stattg – List Target Status

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

## stattg

### Examples

```
| nvdms 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 TME 10 Software Distribution on Your Workstation

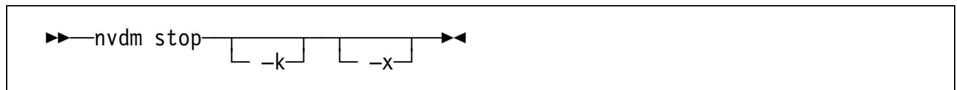
Use the **stop** command to stop TME 10 Software Distribution on the workstation at which the command is executed. A parameter specifies which of the two types of stop is performed.

- 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:

```
nvdmsvr myself
```

Note that no server in the network can be named MYSELF.

**Syntax****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**

```
nvdmsvr 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

```

▶▶—nvdms trace—┐ ┌—-lN comp. name—┐ ┌—-v comp. name—┐▶▶

```

### Parameters

#### **-lN** *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 188.

## trace

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
rhm	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 -12 co -13 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

---

### 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 (C:\softdist\findapi) is located at the server.

### Related Information

The troff command.

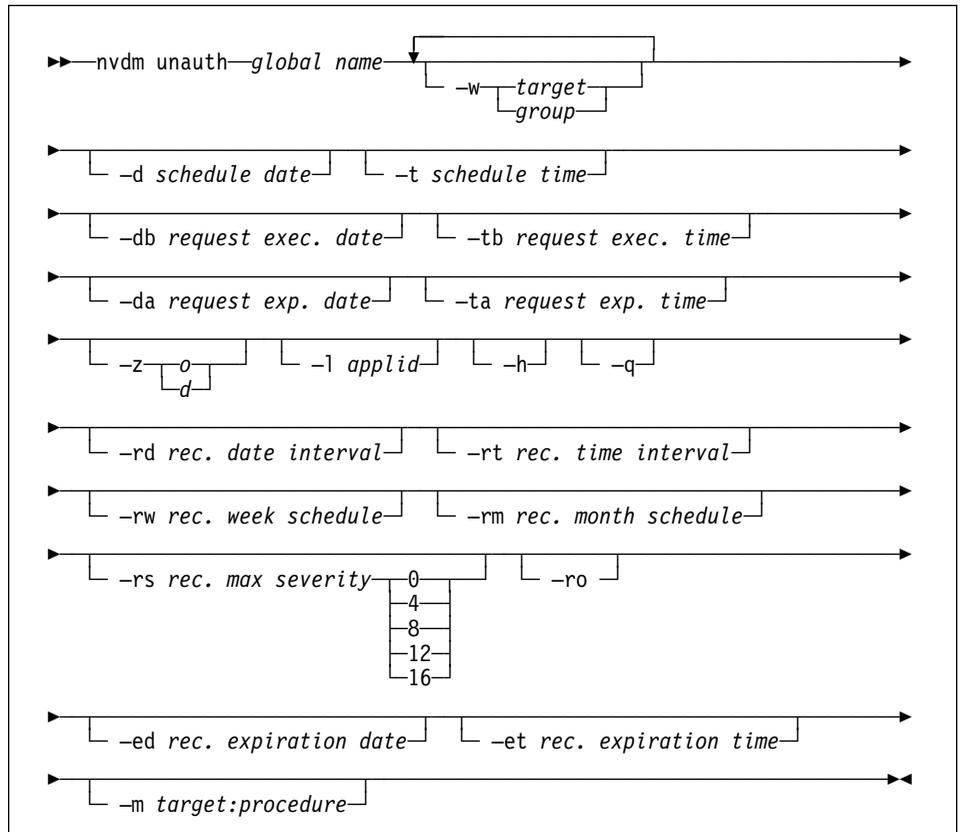
The section of the *User's Guide* about controlling messages and traces.

## 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 12.

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 11.

**-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 12.

**-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.

## unauth

**-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 15.

## Examples

```
nvdn unauth EURO.WORDPROC.* -w FREDSWS
```

This command removes the authorization for FREDSWS to view or use any of the catalog entries relating to the Eurotravel word processor.

## Related Information

The auth and ls commands.

## unbid – Unpack the components of a change file

Use the **unbid** 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 **unbid** command is not effective for objects with their TYPE set to:

- REMOTE\_FILE
- REMOTE\_FILE\_WITH\_TOKENS
- DIRECTORY
- REMOTE\_IMAGE
- REMOTE\_IMAGE\_PMP

The **unbid** 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 **unbid** command only with generic change files. Only a user defined as root to the operating system can unbuild installp change files.

The component files are restored with the following attributes:

**Owner**      The user who submitted the **unbid** command.

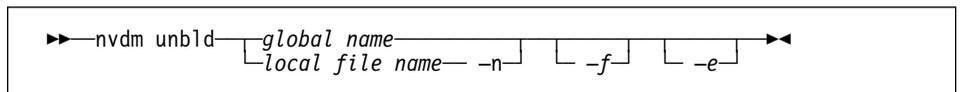
**Group**      FNDADMN

### 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



# unbld

## Parameters

### *global name*

The global name of the change file to be unbuilt. 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

```
nvdn 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:          c:\user\lpp\file.a
TARGET NAME:          c:\user\file.a
TYPE:                 FILE
ACTION:               COPY

OBJECT:
SOURCE NAME:          c:\user\lpp\file.b
TARGET NAME:          c:\user\file.b
TYPE:                 FILE
ACTION:               COPY

OBJECT:
SOURCE NAME:          file.c
TARGET NAME:          c:\user\file.c
TYPE:                 FILE
ACTION:               COPY

OBJECT:
SOURCE NAME:          c:\user\lpp\file.d
SOURCE AT INSTALL:   c:\mount\usr\lpp\file.d
TARGET NAME:          c:\user\file.d
TYPE:                 REMOTE FILE
ACTION:               COPY
```

The **unbld** command unpacks `file.a` and `file.b` in `c:\user\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

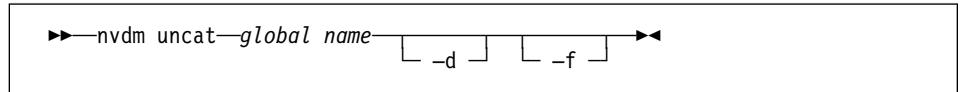
---

### 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 12.

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

```
nvdm 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

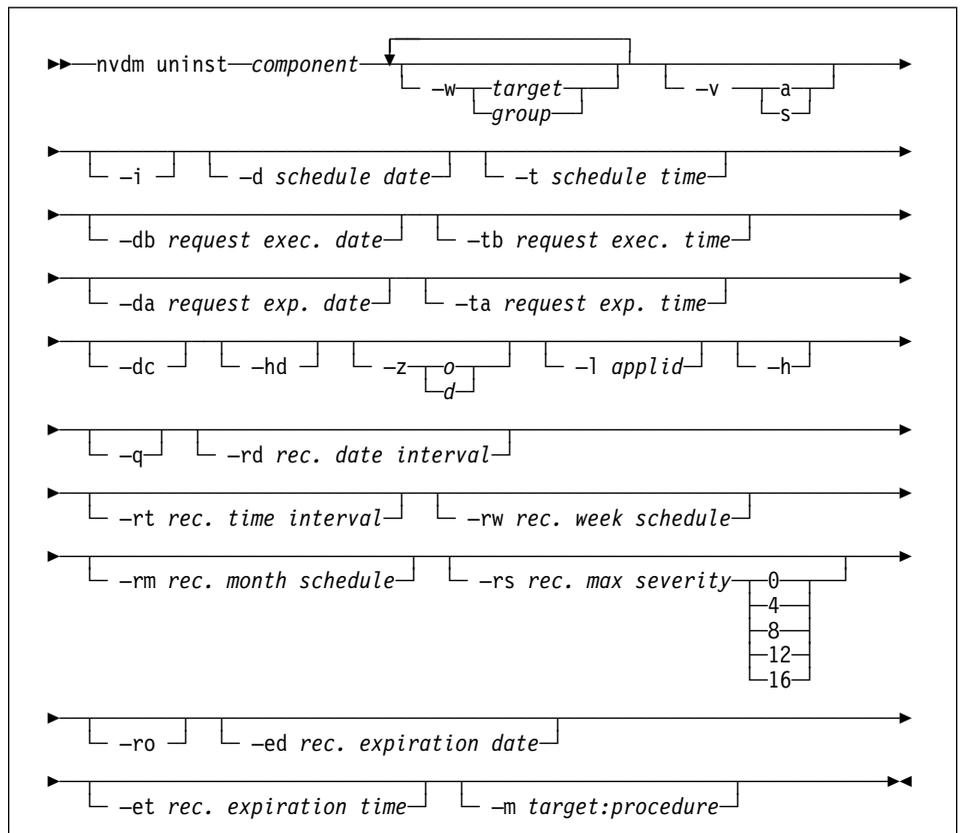
Use the **uninst** command to remove a previously installed component from a target. The component may have originally been built from the installation of multiple change files, all of which are removed.

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 11.

This parameter cannot be used locally by mobile clients.

### **-v** *a|s*

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 12.

### **-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.

**-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 15.

This parameter cannot be used locally by mobile clients.

## Examples

```
nvdms uninst EURO.WORDPROC -w FREDSWS -va -d "3/1/00" -t "2:00"
```

This command uninstalls the component EURO.WORDPROC from target FREDSWS at the time and date specified, if the system is quiesced at that time.

## Related Information

The inst and rem commands.

## upd

---

### 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.

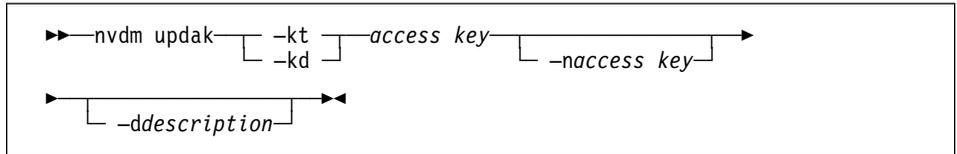
### Related Information

The cat command.

## updak – 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

```
nvdms 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.

## updbs

---

### updbs – Update Configuration

Use the **updbs** 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



### 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

```
nvdms updbs -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.

---

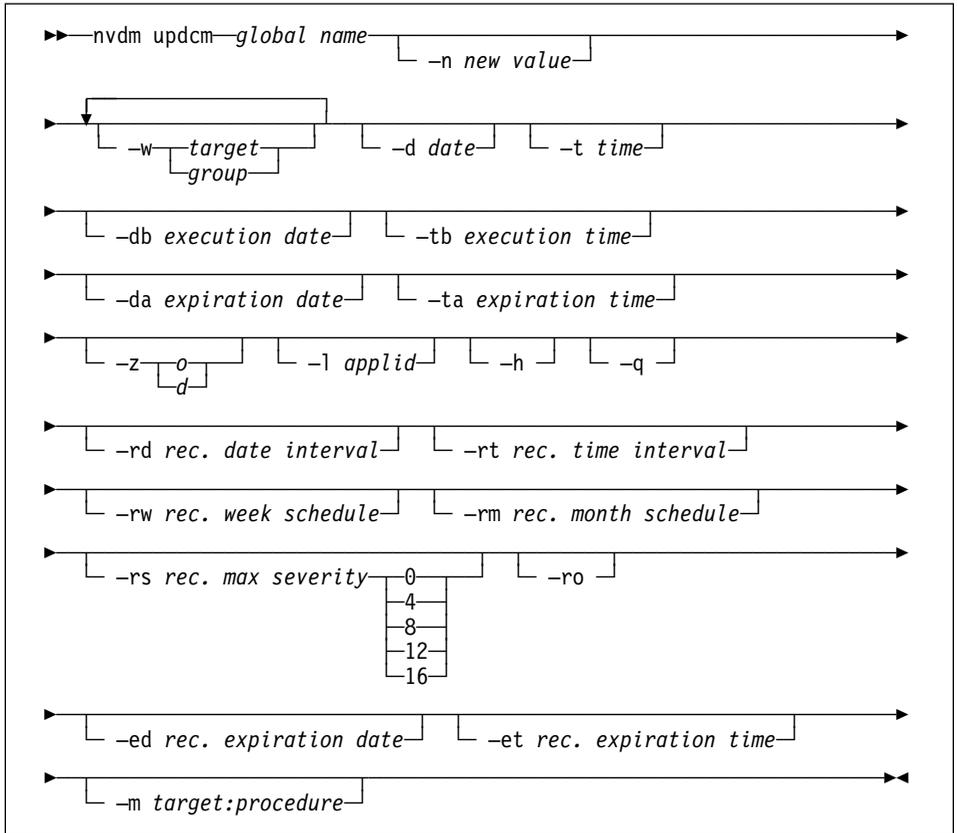
## updcm – Update Change Control Status

Use the **updcm** 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 **updcm** command can be used to modify status information.
- When change control statuses at the server do not reflect the actual situation at targets. **updcm** can be used in such a situation to set the status correctly.

The **updcm** 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 12. 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:

<b>Acronym</b>	<b>Stands For</b>
activated	activation for the target (see below)
available	available
discovered	discovered
distributed	distributed
inra	installed, not removable, active
inri	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 11.

**-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 12.

**-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** *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.

**-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:

<b>0</b>	Successful
<b>4</b>	Warning
<b>8</b>	Error
<b>12</b>	Failed
<b>16</b>	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.

- 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 15.

## Examples

```
nvdcm updcm 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".

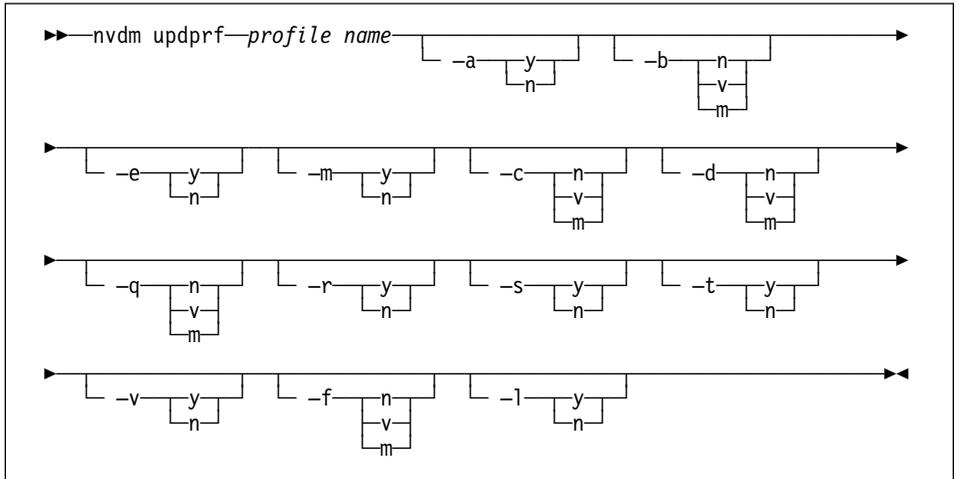
## Related Information

The delcm, lscm, and vercm commands.

## 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.

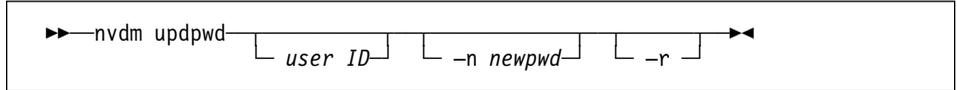
## Related Information

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
```

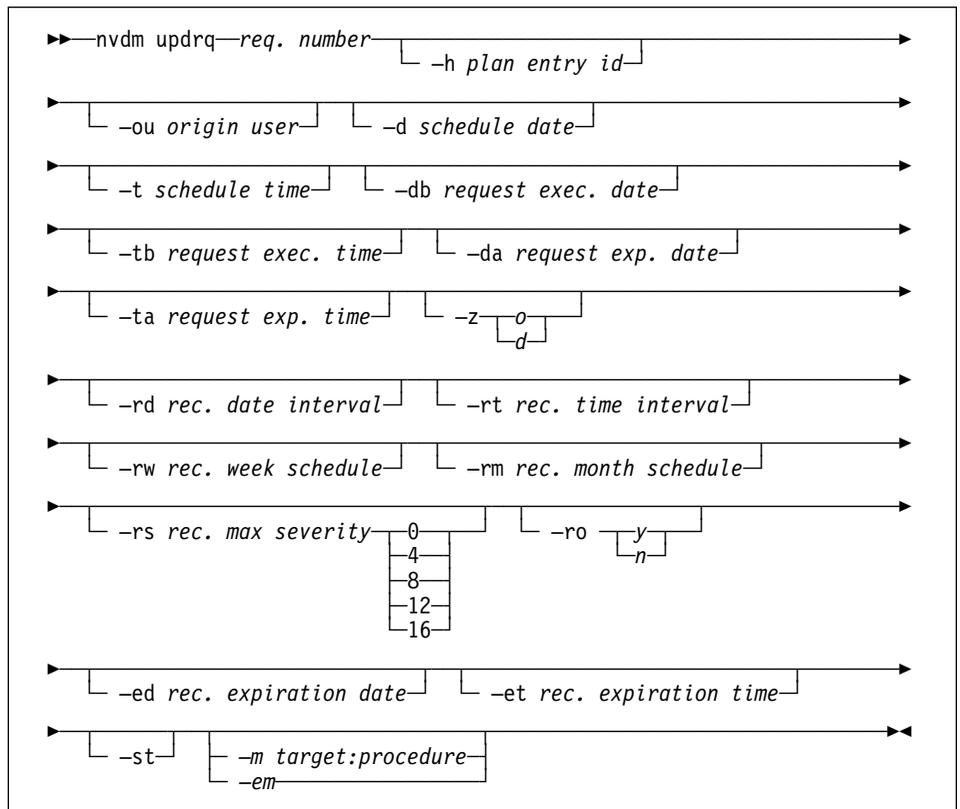
This command updates the password for user JANEDOE.

## 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 12.

**-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 15.

**-em**

Cancel the procedure execution.

## Examples

```
nvdms 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.

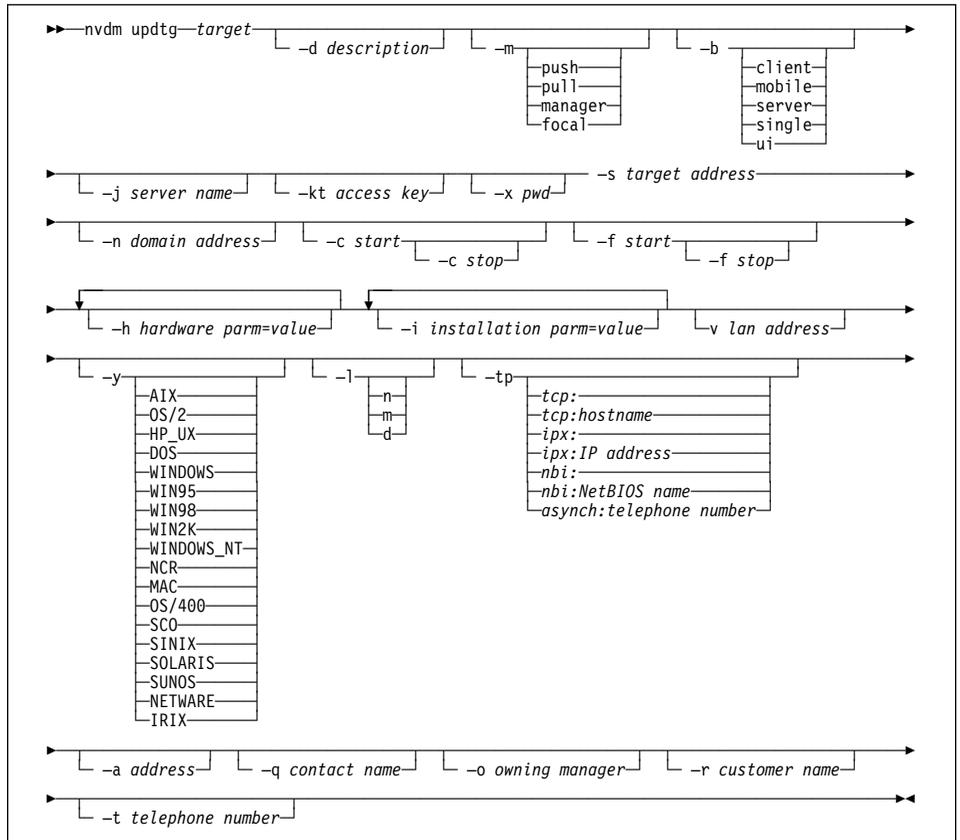
If there are pending requests for the target, you cannot change the name of the target, the server name (**-j** parameter), target address (**-s** parameter), and domain address (**-n** parameter).

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:

<b>push</b>	Push mode target
<b>pull</b>	Pull mode target
<b>manager</b>	Manager node
<b>focal</b>	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:

<b>client</b>	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 <b>-j</b> option.
<b>server</b>	The target is a server.
<b>single</b>	The target is a single node.
<b>ui</b>	The target is user interface only.
<b>mobile</b>	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:

<b>n</b>	Normal (the default)
<b>m</b>	Minimal
<b>d</b>	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.

**ipx:** The transmission protocol is IPX.

**ipx:IP address**

The transmission protocol is IPX, with the IP address you specify.

**nbi:** The transmission protocol is NetBIOS.

**nbi:IP address**

The transmission protocol is NetBIOS, with the NetBIOS name 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.

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

```
nvdms updtg FREDSW -m pull -d "Fred's Workstation (pull mode)"
```

This command updates the local target FREDSW. It becomes a pull mode target. This change is reflected in a new description for the target.

```
nvdms updtg HOST -m remote -d "The host" -r "Head Office"
-q "Mr P.C. Dixon" -t "071 230 1212"
```

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.

**updtg**

**Related Information**

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



# updusr

## **-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:

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 FRED5WS
```

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 FRED5WS.

## Related Information

The addusr, delusr, and lsusr commands.

---

## vercm – Verify Change Control Status

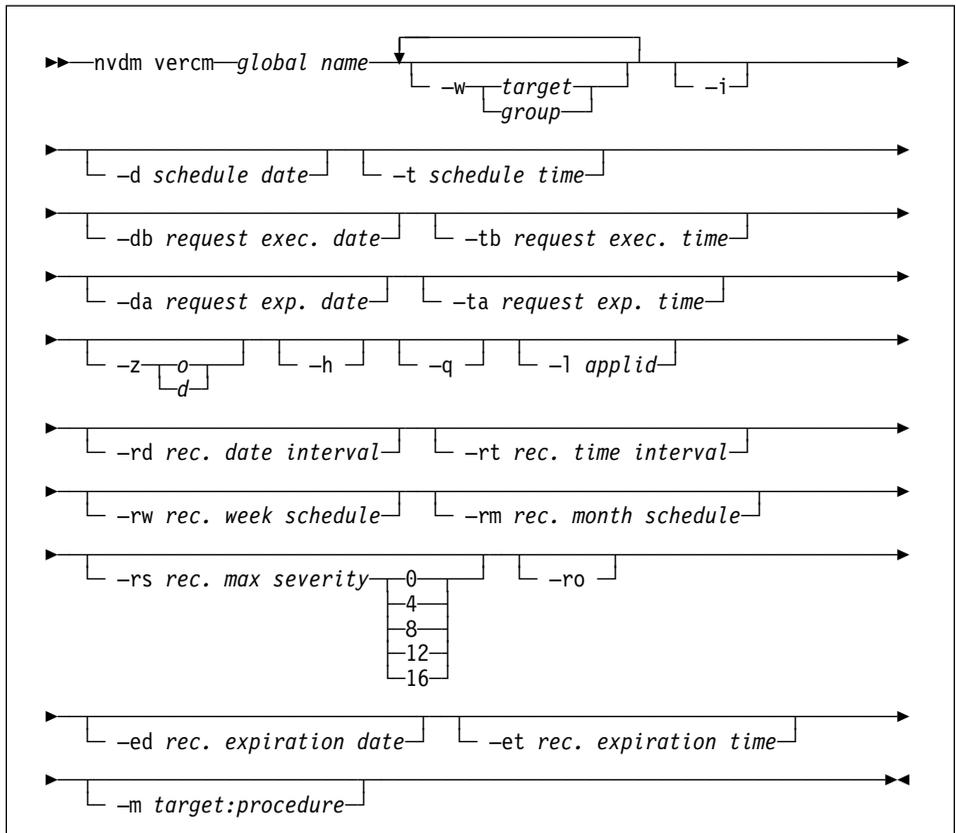
Use the **vercm** command to verify:

- The consistency between the change control status of one or more install 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 install 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 11.

- 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 12.

### **-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 15.

## 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.

---

## REGISTER - CID Software Preparation Command

Run the **REGISTER** command to create an entry in the CID Software Library window for a new software product from its *ADF*.

Enter the following command:

```
REGISTER <SwLib_path>\SoftwareId.ADF
```

Where:

### **SwLib\_path**

Is the full path of the software product ADF

### **SoftwareId.ADF**

Is the file name of the software product ADF

When you open the Software Library window of CID Software Preparation, you find the icon of the corresponding product.

---

## DiskCamera Command Line Operations

Use DiskCamera to create a change file for installing non-CID applications.

This utility automatically creates a change file profile for software that you install on your workstation.

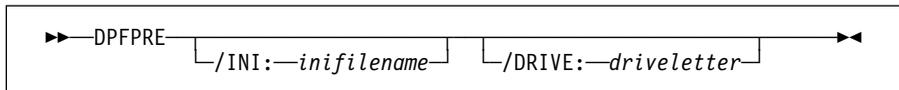
It creates a profile by:

- Taking a picture of the drive you specify and of the configuration files
- Taking a picture of the same drive after you install the application and of the configuration files
- Comparing the two pictures

You can then use the profile to create the change file and install the application at the other targets by means of TME 10 Software Distribution.

To prepare a software package at the preparation site using DiskCamera, perform the following steps:

1. Enter the DPFPRE command with the following syntax:



where:

- *inifilename* is composed of two sections. The **[files to exclude in...]** and the **[files to monitor in...]** section.
  - **[files to exclude in...]** lists the files you do not want to monitor. For example, if you are using DiskCamera for OS/2, you can specify the files to exclude as follows:
 

```
$(boot):\OS2\SYSTEM\swapper.dat
$(prod)
```
  - **[files to monitor in...]** lists the files you want to monitor. For example, if you are using DiskCamera for OS/2, you can specify the files to monitor as follows:
 

```
$(boot):\config.sys
$(boot):\startup.cmd
$(boot):\os2\os2.ini
$(boot):\os2\os2sys.ini
$(boot):\os2init.cmd
```

If you do not specify a name, the default is *BIN\DSKCAM.INI*.

For OS/2, *DSKCAM.INI* lists *OS2INIT.CMD*, *OS2SYS.INI*, *OS2.INI*, *STARTUP.CMD*, and *CONFIG.SYS* to be monitored.

For Windows, it lists *CONFIG.SYS*, *AUTOEXEC.BAT*, and all Windows *.INI* files to be monitored. In either case, you can change the list of files to be monitored by editing the *DSKCAM.INI* file. See “Examples of the DISKCAM.INI Files on the Windows Platforms” on page 238.

- *driveletter* is the name of the drive on which you want to install the application. The procedure monitors this drive. The system directory is also monitored if it is located in a different drive.

The DPFPRE command records the current status of the specified drive and of the configuration files by taking a picture of them. It creates the wrk system directory with the copy of the file to monitor.

2. Install the application on the drive you specified in the *driveletter* option of the DPFPRE command.

DiskCamera monitors each workplace object (icon, folder, launch pad, etc.) on the current desktop only if it was created with an object ID.

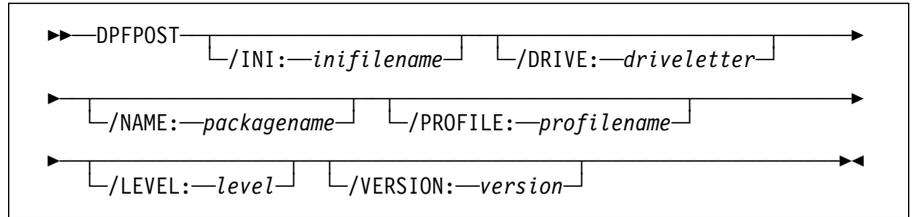
On Windows 95, DiskCamera cannot monitor direct creation of an icon on the desktop or in the Programs folder. To bypass this limitation, follow these steps:

- a. From the Windows 95 task bar, select **Start** and click on it with the right mouse button.
- b. Select **Open** and open the Programs folder.

- c. Create a new folder with the product name.
- d. Under this folder, create a new icon and link it to the related product executable file.

Then take the *after picture*.

3. Enter the DPFPPOST command with the following syntax:



where:

- *inifilename* is the name of the file that lists all the configuration files that have been monitored. If you do not specify a name, the default is *DSKCAM.INI*.
- *driveletter* is the name of the drive on which you installed the application. The procedure monitors this drive. The system directory is also monitored if it is located in a different drive.
- *packagename* is used to define:
  - The global name of the change file in the profile as follows:  
GLOBAL NAME: <packagename>.REF.<level>.<version>
  - The destination directory for the DPFPUPD.MOD file as follows:  
<drive>:\<packagename>\DPFPUPD.MOD  
If you do not specify /NAME:, the default value is DSKCAM.
  - *profilename* is the name of the change file profile. It contains all the information to replicate the application installation.
  - *level* is the level of the change file that you create with the change file profile. The default value is 100.
  - *version* is a text of your choice. It will be the third part of the change file's name in the software distribution catalog. The default value is 1.

The DPFPPOST command does the following:

- a. Records the current status of the drive on which you installed the product by taking a picture of it.
- b. Compares the new configuration with the old one.
- c. Creates a modification file with a list of the modifications made to the configuration files during the installation of the application. The default name of the modification file is DPFPUPD.MOD. See “Modification File Statements” on page 239 for more information.

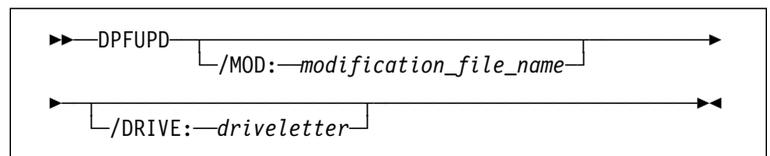
This file is located in the DSKCAM directory of the drive on which you installed the application.

If you are in a Microsoft Windows environment, DiskCamera also creates the DELTAREG file. The DELTAREG file contains the list of modifications made to your system registry. It is stored in the same directory as the modification file.

- d. Creates a change file profile that specifies:
- The files to install and their location.
  - The actions to run after the installation of these files.
  - The DPFUPD command (in the post-install section) that performs all the changes to the configuration files at the targets during the installation of a change file created by DiskCamera.

If you are in a Microsoft Windows environment, DiskCamera uses the modification file and the DELTAREG file as input files to make the changes. Before running the DPFUPD command, make sure that the DELTAREG file and the modification file are stored in the same directory. You cannot specify the DELTAREG file as an input parameter of the DiskCamera commands.

If you are in an OS/2 environment, DiskCamera uses the modification file only as an input file to make the changes. You can also run this utility from a command prompt. Its syntax is:



where:

- */MOD: modification\_file\_name* is the fully qualified name of the modification file.
- */DRIVE: driveletter* specifies a target directory to be used by the \$(TargetDir) macro in the modification file. If this parameter is not specified, then \$(TargetDir) has an empty value.

The default name of the change file profile is DSKCAM.PRO. The profile is located in the BIN directory of the drive in which DiskCamera commands are installed.

You can use this profile to create the change file and to replicate the application installation on other workstations.

## DiskCamera Error Codes

The following return codes are given by DiskCamera when an error occurs. These error codes are logged in the DSKCAM.LOG file that is stored in the TME 10 Software Distribution installation directory.

<u>VALUE</u>	<u>MEANING</u>
16	An error has been detected parsing the command line parameter: %d
18	Failure allocating memory
19	An error has been detected attempting to retrieve the syslevel files: %s, %s.
20	An error has been detected attempting to retrieve the boot drive information
21	An error has been detected opening the %s.INI file
22	Syntax error in the %s.INI file in the %s line
23	An unexpected error has been detected
24	An error has been detected attempting to open the %s file
25	An error has been detected attempting to retrieve the windows system directory
26	An error has been detected writing to the %s file
27	Invalid handle accessing the registry
28	An error has been detected reading to the %s file
29	A line in the text field exceeds the maximum length
31	The %s file is too big to be processed
32	The %s group file could not be found
33	The %s group information could not be addressed
34	An error has been detected updating the %s.MOD file
35	An invalid statement has been detected processing a system file
36	An invalid add command has been detected
37	A duplicate line has been detected processing a system file
38	The system cannot determine the right position of the add token statement
39	An invalid parameter has been detected in the SET statement
40	MODE statement invalid
41	The MODE statement parameter is invalid
42	Command valid only in the DOS5 environment
43	Command valid only in the DOS4 environment
44	The %s file was not found
45	Invalid %s argument
46	An error has been detected accessing the % modification file
47	The %s modification file is empty
48	The %s target drive is invalid
49	A syntax error has been detected in the .MOD file. A section without statement has been detected
50	A syntax error has been detected in the .MOD file. A section is expected
51	The system cannot remove the %s directory
52	The processing command is duplicate
53	An error has been detected writing a profile string to the %s file

54	A syntax error has been detected in the .MOD file. The %s statement is not valid.
55	A syntax error has been detected in the .MOD file. The %s token was not found.
56	The token specified already exists.
57	A syntax error has been detected in the .MOD file. The DOS command %s is invalid.
58	A syntax error has been detected in the .MOD file. The command %s is not supported at this processing stage.
59	A syntax error has been detected in the .MOD file. The %s sub-command does not match a valid command.
60	A syntax error has been detected in the .MOD file. The %s DOS command cannot be processed at this stage.
61	The system cannot delete the %s file
62	A syntax error has been detected in the .MOD file. Invalid %s command.
63	Invalid file name parameter.
64	A syntax error has been detected in the .MOD file. Invalid value in the %s position field.
65	Unexpected end of file parsing the .MOD file
66	A syntax error has been detected in the .MOD file. Invalid %s token
67	An error has been detected opening the registry for the %s sub-key.
68	Error reading the %s string value from the Registry/.INI file
70	A command line error has been detected. The %s option has already been defined.
71	A command line error has been detected. The %s option has no value.
72	A command line error has been detected. The %s mandatory option has not been specified.
73	A command line error has been detected. The %s option cannot have an associated value.
74	An error has been detected attempting to create the %s directory.
75	Expected a directory, but the %s path refers to a file.
79	Unable to get information for the %s file.
80	The disk is full.
81	Unable to get information for the %s folder.
83	An error has been detected attempting to copy the %s file to %s file

## Examples of the DISKCAM.INI Files on the Windows Platforms

The DISKCAM.INI file is composed of two sections, the **[files to exclude in...]** and the **[files to monitor in...]** sections.

The **[files to exclude in...]** section lists the files you do not want to monitor.

The **[files to monitor in...]** section lists the files you want to monitor.

The default DISKCAM.INI files for the WINDOWS 95, WINDOWS 3.1 and WINDOWS NT platforms follow:

```

[filestoexcludeinWindows95}
$(PROD)=
$(WINDIR)\WIN386.SWP
$(WINDIR)\*.LNK
$(WINDIR)\*.*
[filestomonitoriesinWindows95}
$(BOOT):\CONFIG.SYS
$(BOOT):\AUTOEXEC.BAT
$(WINDIR)\SYSTEM.DAT
$(WINDIR)\USER.DAT
$(WINDIR)\*.INI

[filestoexcludeinWindows31}
$(PROD)=
$(WINDIR)\WIN386.SWP
[filestomonitoriesinWindows31}
$(BOOT):\CONFIG.SYS
$(BOOT):\AUTOEXEC.BAT
$(WINDIR)\REG.DAT
$(WINDIR)\*.INI

[filestoexcludeinWindowsNT}
$(PROD)
[filestomonitorinWindowsNT}
$(BOOT):\CONFIG.SYS
$(BOOT):\AUTOEXEC.BAT
$(WINDIR)\*.INI

```

where:

- \$(PROD) is the TME 10 Software Distribution product directory. The DISKCAMERA application is installed in the \$(PROD)\BIN directory.
- \$(WINDIR) is the Windows directory.
- \$(BOOT) is the boot drive name.

## Modification File Statements

The following describes the syntax of the statements that you can use in the modification file.

### [system\_file\_name]

Identifies the system file, text file, or directory to be modified. If you do not insert a path name for the CONFIG.SYS, AUTOEXEC.BAT, or .INI files, a default path is selected. VMFCLEAR The startup drive path is used for the CONFIG.SYS and AUTOEXEC.BAT files, and the Microsoft Windows installation directory is used for the .INI file. If you want to modify the file EXCEL.INI under C:\EXCEL, use:

```
[C:\EXCEL\EXCEL.INI]
```

The left and right square brackets are mandatory. All statements that follow the specified [system\_file\_name] statement are applied to the identified system file until a new [system\_file\_name] statement is found.

## Modification File Statements

Example:

```
[CONFIG.SYS]
```

All statements following this one are applied to the system file CONFIG.SYS until another [*system\_file\_name*] statement is found.

### **InsertCommand**(*cmd\_name*,*cmd\_attrib*, *position*)

*cmd\_line*

Modifies or adds a system file line that contains the DOS command *cmd\_name*. The *cmd\_name*,*cmd\_attrib* combination is used as a key to identify a line of the system file to modify. In processing the InsertCommand statement, the utility looks for a line that contains the combination and, if found, changes that line with the *cmd\_line* value; otherwise, it adds *cmd\_line* to the file. If the system file contains more than one line with the *cmd\_name*,*cmd\_attrib* combination, no lines are updated and an error occurs. To update such lines, you can use the AddLine statement.

Example:

```
InsertCommand(DEVICE,SMARTDRV.SYS,BOTTOM)
DEVICE=C:\WIN30\SMARTDRV.SYS 2048
```

Only the following *cmd\_name* values require a *cmd\_attrib* value: DEVICE, DEVICEHIGH, and INSTALL for the CONFIG.SYS file, and SET, MODE, and LH/LOADHIGH for the DOS batch files (see Table 11 on page 246).

### **DeleteCommand**(*cmd\_name*,*cmd\_attrib*)

Deletes a system file line that contains the specified *cmd\_name* value.

Example:

```
DeleteCommand(DEVICE,SMARTDRV.SYS)
```

### **InsertToken**(*cmd\_name*, *cmd\_attrib*,*position*)

*token*

Modifies or adds the specified token to a line in a system file that contains the specified *cmd\_name* value.

This statement can be used for *cmd\_name* values of the form:

```
KEY=TOKEN;TOKEN; ... TOKEN;
```

The *cmd\_name* values that belong to this category are those used for the DOS batch files SET, APPEND, and PATH.

The *token* value must be a valid DOS path name. Only one token defining the path of one file or directory can be included in each statement. You cannot concatenate tokens with the semicolon (;).

Example:

```
InsertToken(SET,INCLUDE,RIGHT)
C:\WINDEV\INCLUDE
```

**DeleteToken**(*cmd\_name*,*cmd\_attrib*)*token*

Deletes the specified token from a line in a system file that contains the specified in *cmd\_name* value.

The *token* value must be a valid DOS path name.

Example:

```
DeleteToken(SET,INCLUDE)
C:\WINDEV\INCLUDE
```

**AddLine**(*position*)*line*

Modifies or adds a line at the specified position in a system file or in a text file.

The field *line* can be any sequence of characters.

Example:

```
AddLine(TOP)
WIN
```

**DeleteLine**()*line*

Deletes a line from a text or system file.

Example:

```
DeleteLine()
WIN
```

**DeleteFile**()

Deletes the file specified in the preceding [*system\_file\_name*] statement.

The file is deleted even if it is read-only.

Example:

```
[\WINDOWS\CONFIG.WIN]
DeleteFile()
```

**DeleteTree**()

Deletes the directory and all its subdirectories specified in the preceding [*system\_file\_name*] statement.

Example:

```
[\WINDOWS\SYSTEM]
DeleteTree()
```

**CreateGroup**(*group\_name*)

Valid only under Microsoft Windows.

Creates the specified group in the Microsoft Windows program manager.

Example:

## Modification File Statements

```
CreateGroup(Microsoft Excel 4.0)
```

### **DeleteGroup**(*group\_name*)

Valid only under Microsoft Windows.

Deletes the specified group in the Microsoft Windows program manager.

Example:

```
DeleteGroup(Microsoft Excel 4.0)
```

### **AddGroupItem**(*group\_name,command\_line,item\_description,working\_dir*)

Valid only under Microsoft Windows.

Adds the specified group item to the group of the program manager.

A group item is an icon that identifies an application. You can click on the icon to execute the application.

Example:

```
AddGroupItem(Microsoft Excel 4.0,$(TargetDir)EXCEL\
excelde.exe,Dialog Editor,$(TargetDir)EXCEL)
```

### **DelGroupItem**(*group\_name,item\_description,*)

Valid only under Microsoft Windows.

Deletes the specified group item to the group of the program manager.

A group item is an icon that identifies an application. You can click on the icon to execute the application.

Example:

```
DelGroupItem(Startup,Start Distribution Client)
```

### **AddUniqueKeyPrf**(*section,key*)

*value*

Valid only under Microsoft Windows.

Adds the unique key value to the specified section for Microsoft Windows.

A unique key is a key that can appear only once in the Microsoft Windows .INI file. Most of the keys are unique. Some keys can be multiple, for example the key device of the SYSTEM.INI file, used to load Microsoft Windows device drives, can appear more than once. Use AddMultipleKeyPrf to add such keys.

Example:

```
AddUniqueKeyPrf(MSWord Text Converters,MSWordDos)
Word for DOS$(c)$
(TargetDir)WINWORD\WORDDOS.CNV$(c)
```

### **DelUniqueKeyPrf**(*section,key*)

*value*

Valid only under Microsoft Windows.

Deletes the unique key value from the specified section for Microsoft Windows.

Example:

```
DelUniqueKeyPrf(MSWord Text Converters,MSWordDos)
Word for DOS$(c) $(TargetDir)WINWORD\WORDDOS.CNV
$(c)
```

### **AddMultipleKeyPrf**(*section,key*)

*value*

Valid only under Microsoft Windows.

Adds the key value to the specified section for Microsoft Windows. This particular key can have multiple occurrences within the section.

Example:

```
AddMultipleKeyPrf(386Enh, device)
device=*vmcpd
```

### **DelMultipleKeyPrf**(*section,key*)

*value*

Valid only under Microsoft Windows.

Deletes the key value from the specified section for Microsoft Windows. This particular key can have multiple occurrences within the section.

Example:

```
DelMultipleKeyPrf(386Enh,device)
device=*vmcpd
```

### **AddTokenPrf**(*section,key*)

*token*

Valid only under Microsoft Windows.

Adds the token value to the specified key for Microsoft Windows.

The statement is useful for the keys of the Microsoft Windows .INI file of the form:

```
KEY=TOKEN;TOKEN; ... TOKEN;
```

Example:

```
AddTokenPrf(windows,load)
MYPROG.EXE
```

### **DelTokenPrf**(*section,key*)

*token*

Valid only under Microsoft Windows.

Deletes the token value from the specified key for Microsoft Windows.

Example:

```
DelTokenPrf(windows,load)
MYPROG.EXE
```

## Modification File Statements

### **XCOPY**(*target*)

Selectively copies a group of files, which can include lower-level subdirectories. Specify the drive, the path, and the file name for the target files. If you do not specify a path, XCOPY starts from the current directory. If you do not specify a file name, XCOPY uses \*.\* as the default value.

XCOPY copies the files in the source directory and in all directories below the starting source directory.

XCOPY preserves the attributes of the source file. For example, a read-only file is copied to another read-only file.

If the specified target path does not exist, XCOPY creates the directory before copying. XCOPY also copies empty source directories.

If the files you want to copy already exist in the target directory, they are overwritten.

Example:

```
[C:\SOURCE\*.*)
XCOPY(D:\TARGET\*.*)
```

### **CreateFolder**(*name,display\_name,icon\_path,icon\_index*)

Create a folder in a Microsoft Windows environment.

Valid only under Windows 95, Windows 98, Windows 2000 or Windows NT 4.0 or later.

*name*

The complete path of the folder.

*display name*

The name of the folder.

*icon path*

The complete path of the resource file that contains the icon.

*icon index*

The icon index in the resource file

Example:

```
CreateFolder($(DESKTOP)\MyFolder,MyFolder,$(WINSYSDIR)\SHELL32.dll,5)
```

This example creates the MyFolder folder on the desktop.

### **RemoveFolder**(*name*)

Remove a folder in a Microsoft Windows environment.

Valid only under Windows 95, Windows 98, Windows 2000 or Windows NT 4.0 or later.

*name*

The complete path of the folder.

Example:

```
RemoveFolder(C:\WINNT\Profiles\CDAMORE\Desktop\MyFolder)
```

This example remove the MyFolder folder from the desktop.

**CreateLink**(*name,path,display\_name,working\_directory,arguments,icon\_path,icon\_index,hot\_key*)

Create a link in a shell windows.

Valid only under Windows 95, Windows 98, Windows 2000 or Windows NT 4.0 or later.

*name*

The complete path of the link file. For example:

```
c:\winnt.\...\my shortcut.lnk
```

*path*

The complete path of the program of which you want to create the link.

```
c:\win32app.\tools\program.exe
```

*display name*

The name with which the shortcut is displayed. It can be empty.

*working directory*

The working directory of the program.

```
c:\win32app.\tools
```

*arguments*

The command line arguments.

```
-f input.file
```

*icon path*

The complete path of the resource file that contains the icon.

*icon index*

The icon index in the resource file

*hot key*

The link's hot key

Example:

```
CreateLink($(DESKTOP)\MyFolder\nvdm.lnk,▶
$(PROD)\bin\nvdm.exe,,$(PROD)\bin,,,0,0)
```

This example creates the nvdm.lnk link.

**RemoveLink**(*name*)

Remove a link in a Microsoft Windows environment.

Valid only under Windows 95, Windows 98, Windows 2000 or Windows NT 4.0 or later.

## Modification File Statements

*name*

The complete path of the link file.

Example:

```
RemoveLink(C:\WINNT\Profiles\CDAMORE\Desktop\MyFolder\nvdm.lnk)
```

This example removes the nvdm.lnk link.

### Rules for the Modification File Statements

- The order of the parameters is mandatory.
- If a specified command does not require an attribute, its field must be empty. For example:  

```
DeleteCommand(FILE,)
```
- The following table shows the command attributes required for each command name.

Command Name	Command Attributes	Example
DEVICE	name.ext	InsertCommand(DEVICE,DISPLAY.SYS,TOP)
DEVICEHIGH	name.ext	InsertCommand(DEVICEHIGH,SETVER.EXE,TOP)
INSTALL	name.ext	InsertCommand(INSTALL,FASTOPEN.EXE,TOP)
SET	variable	InsertCommand(SET,TMP,TOP)
MODE	port	InsertCommand(MODE,CON,TOP)
LH/LOADHIGH	name.ext	InsertCommand(LH,DOSKEY.COM,TOP)

- Statements cannot contain a blank line or control characters other than line feed, carriage return, and EOF.
- The LH and LOADHIGH commands are considered as the same command.
- The commands SET PATH and PATH are treated as the same command on the DOS workstation. If you use one command in the modification file, and the other command is already present in the file, the change affects the one already there. For example:

If the CONFIG.SYS file on the workstation contains the line:

```
PATH C:\PIPP0\PLUTO;
```

the statement

```
InsertToken(SET,PATH,RIGHT)  
C:\MYDIR
```

causes the existing statement in the CONFIG.SYS file to change as follows:

```
PATH C:\PIPP0\PLUTO;C:\MYDIR
```

- Macros can be used in the modification file. Specific values are substituted for them when the modification file is processed.

For example, if you want to delete the file WINVER under the Microsoft Windows installation directory, enter:

```
[$(WindowsDir)\WINVER]  
DeleteFile()
```

Possible macros are:

\$(TargetDir)	TME 10 Software Distribution target directory
\$(WindowsDir)	Microsoft Windows installation directory
\$(c)	Comma

Macros are allowed as follows:

- \$(TargetDir) and \$(WindowsDir) when the field can contain a file name.
- \$(c) in the fields followed by comma separators.



---

## 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.

---

### 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 253, and “Dynamic Section Keywords” on page 265 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 250.

## The Structure of a Change File Profile

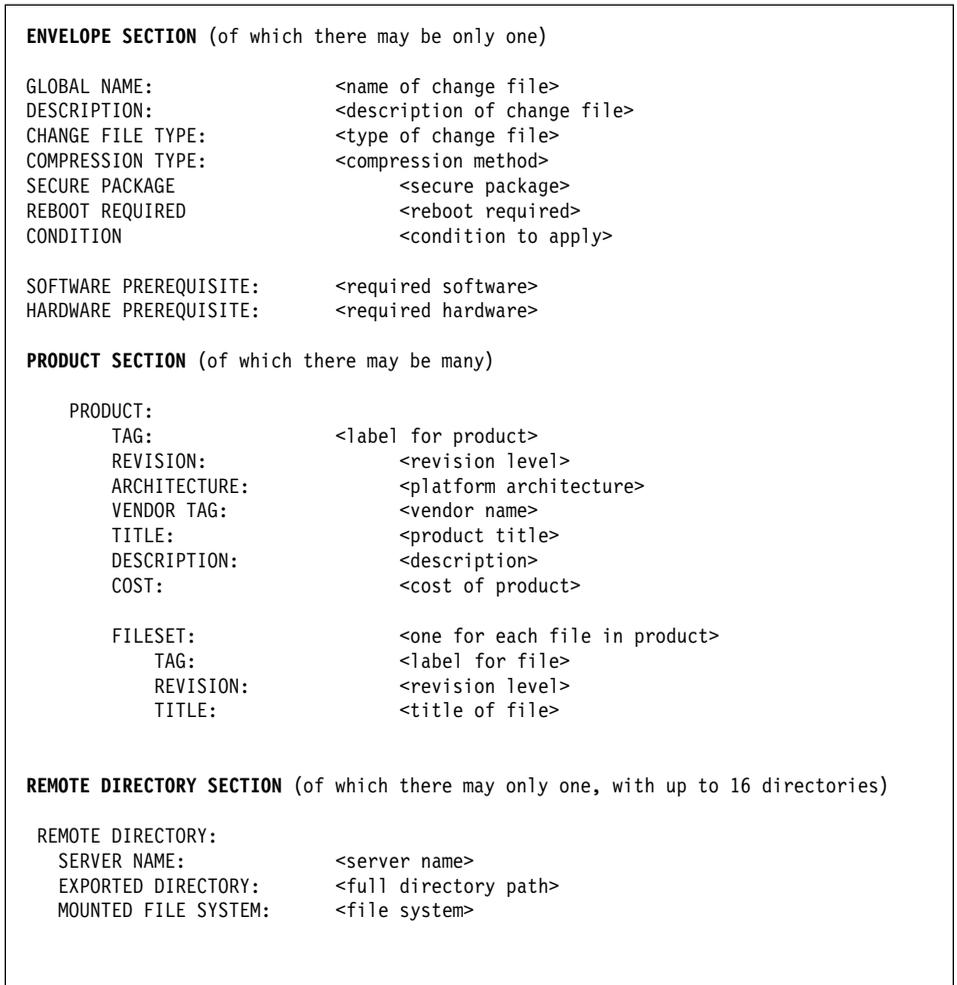


Figure 1 (Part 1 of 2). The structure of a Change File profile

<b>DYNAMIC SECTION</b> (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>
<b>OBJECT SECTION</b> (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.

## Change–File Profile Keywords

---

### 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.

### 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 265). 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.

## Change-File Profile Keywords

**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.

## 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.

## Change-File Profile Keywords

**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.

### **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*
- *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>
```

## Change-File Profile Keywords

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.

### **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:

## Change-File Profile Keywords

- 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:  c:\velka\source\aix\bin  
  MOUNTED FILE SYSTEM: c:\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:      c:\u\user\dir2
TARGET NAME:      c:\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 265.

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 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.

### 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

## Change-File Profile Keywords

- 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.

### 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

#### 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.

## Change-File Profile Keywords

### 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]
```

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:      c:\usr\lpp\netviewdm\pre-install

DYNAMIC SECTION:  small_machine
CONDITION:        DRIVE_SIZE < 180M
PRE-INSTALL:      c:\usr\lpp\netviewdm\pre-install-small

OBJECT:
DYNAMIC SECTION:  small_machine
SOURCE NAME:      c:\user\lpp\netviewdm\file_small
TYPE:             FILE
ACTION:           COPY

OBJECT:
DYNAMIC SECTION:  big_machine
SOURCE NAME:      c:\user\lpp\netviewdm\file_big
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.

## Examples of TME 10 Software Distribution Change File Profiles

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 253. A condition must be specified.

### SOFTWARE PREREQUISITE

An optional software prerequisite, expressed in the format described in 253.

### REMOTE DIRECTORY

An optional statement that defines the remote location where software is stored, as described in "Remote Directory Section Keywords" on page 260.

### PRE-, POST- ACTIONS

Optional statements that define procedures or commands to be executed before or after the change control operation, as described on page 257.

---

## 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:           c:\user\wordproc\fix\*
    TARGET NAME:           $(EuroWord)\*
    TYPE:                  FILE
    ACTION:                COPY

OBJECT:
    SOURCE NAME:           c:\user\wordproc\menus
    TARGET NAME:           $(EuroWord)\menus
    TYPE:                  FILE
    ACTION:                COPY

OBJECT:
    SOURCE NAME:           c:\u\user\dir2
    TARGET NAME:           c:\u\user\dir2
    TYPE:                  DIRECTORY
    ACTION:                CREATE

OBJECT:
    SOURCE NAME:           c:\u\nfs\remote_file
    TARGET NAME:           c:\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 `c:\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 `c:\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.

## Creating CID Software Profiles

```
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: c:\user\lpp\netviewdm
MOUNTED FILE SYSTEM: c:\home\source
SERVER NAME:        srv03
MOUNT OPTIONS:      -b 4092
EXPORT OPTIONS:     -ro

REMOTE DIRECTORY:
EXPORTED DIRECTORY: c:\velka\source\aix\bin
MOUNTED FILE SYSTEM: c:\home\r2build\bin
MOUNT OPTIONS:      -b 4092
EXPORT OPTIONS:     -ro

OBJECT:
REMOTE SOURCE NAME: c:\home\source\*
TARGET NAME:        $(NetPlan)\*
TYPE:               REMOTE FILE
ACTION:             COPY

OBJECT:
REMOTE SOURCE NAME: c:\home\r2build\bin
TARGET NAME:        $(NetPlan)\menus
TYPE:               REMOTE FILE
ACTION:             COPY
```

Figure 3. Example of a TME 10 Software Distribution Change File profile

---

## Creating CID Software Profiles

With DOSCID and OS2CID software objects, you install, in unattended mode, a CID product carrying the name and the parameters of the product-supplied installation program.

The following list contains each of the commands allowed in the CID software profile:

- GLOBAL NAME (see 269) (required)
- DESCRIPTION (see 269)
- LOCAL NAME (see 269)
- CHANGE FILE TYPE (see 269) (required)
- PRE- and POST- Scripts (see 270)
- PREREQ and POSTREQ Commands (see 270)
- SOFTWARE PREREQUISITE (see 270)
- HARDWARE PREREQUISITE (see 271)
- DEFAULT TOKEN (see 271)
- INSTALL PROGRAM (see 273) (required)
- BACKUP PROGRAM (see 273)
- REMOVE PROGRAM (see 274)
- ACCEPT PROGRAM (see 274)
- UNINSTALL PROGRAM (see 274)
- MAINTENANCE SYSTEM (see 275) (only on OS/2)

For more details, see Example of CID Software Profile (see 278) .

## GLOBAL NAME Keyword

Global name of the change file to be built. See Entering Global File Names (see 12).

An example is:

```
GLOBAL NAME:      EURO.WORDPROC.FIX.3.F0002.US
```

## DESCRIPTION Keyword

Description of the software object. An example is:

```
DESCRIPTION:      Fix to bugs F0001 and F0002
```

## LOCAL NAME Keyword

Local name for the change file when it is built. An example is:

```
LOCAL NAME:      C:\REPOS\EURONET.CF
```

## CHANGE FILE TYPE

The type of software object. This keyword must be entered in uppercase.

The allowed values are:

```
|      DOSCID (for existing Change Files only)  
|      OS2CID  
|      WINCID (for existing Change Files only)
```

An example is:

## Creating CID Software Profiles

CHANGE FILE TYPE: OS2CID

### PRE- and POST- Scripts

You can run these 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**

An example is:

PRE-INSTALL: PREP.COMD  
POST-INSTALL: CLEANUP.COMD

### PREREQ and POSTREQ Commands

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.

Only one command can be written in the profile. For more commands, a script must be supplied.

PREREQ COMMAND: PRECHECK.COMD  
POSTREQ COMMAND: POSTCHECK.COMD

### SOFTWARE PREREQUISITE Keyword

Global file name of prerequisite software object, which must be already installed. You can enter several software prerequisites.

An example is:

SOFTWARE PREREQUISITE: EURO.WORDPROC.UPD.2.3.US

## HARDWARE PREREQUISITE Keyword

You can specify the hardware with which a target must be equipped before the change file can be installed. The hardware prerequisite can have one of the following forms:

```

name = value
name != value
name < value
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. An example is:

```

HARDWARE PREREQUISITE:    mem > 6000
HARDWARE PREREQUISITE:    monitor = SVGA
    
```

## DEFAULT TOKEN Keyword

Token to be defined during installation of the software object, if not already defined.

Tokens are short pieces of text, such as directory names, that can be referred to from within a change file using a single word (the token name). This takes the form:

```
token=value
```

You can define a default value for a token in a software object. If the token is not defined on a target when the installation takes place, it is assigned that default value.

Token substitution can also be used to amend the contents of scripts that are part of the installation. It is optional for each file in the change file whether or not the installation process should include scanning the file and converting any tokens found to their definitions on the target.

You can also use tokens that are already defined at the target. Their values are not changed when you install the software object. To list these tokens, see *Defined Tokens* (see 271).

*Defined Tokens:* You cannot change the value of the following tokens. They are created automatically when a target is defined.

Token	Meaning
<b>\$(TARGET)</b>	Name of the target. It is composed in the following way: <pre>&lt;SystemName&gt;.&lt;Protocol&gt;.&lt;NetworkAddress&gt;</pre>

## Creating CID Software Profiles

<b>\$(SYSTEMNAME)</b>	Name of the system. It is the first part of the name of the target.
<b>\$(SERVER)</b>	Name of the distribution server for that target.
<b>\$(CMSVCE)</b>	Name of the service area directory of the distribution client. This directory is defined in the base configuration file (NVDM.CFG).  This is the name of a directory on the target which is used to store partially completed change control operations.
<b>\$(CMBKUP)</b>	Name of the backup area directory of the distribution client. This directory is defined in the base configuration file (NVDM.CFG).  This is the name of the directory on the target where backups of removable installations are kept.
<b>\$(CMWORK)</b>	Name of the work area directory of the distribution client.  This is the name of a directory on the target that is used for temporary files created while change control is being performed. It is defined in the base configuration file (NVDM.CFG).
<b>\$(REPOSITORY)</b>	Name of the Repository  This is the directory on the target where cataloged files are stored when they do not have any other location specified. This directory is defined in the base configuration file (NVDM.CFG).

In addition to the previous tokens, the following built-in tokens can appear within a software object. To maintain their mnemonic meaning, their value is automatically refreshed each time a request is executed, (e.g. FREEDRIVE tokens will always have an available drive value).

<b>BOOTDRIVE</b>	Refers to the current boot drive (for example <i>D:</i> ).
<b>FREEDRIVE1</b>	Refers to the first available drive (for example <i>Z:</i> ).
<b>FREEDRIVE2</b>	Refers to the second available drive (for example <i>Y:</i> ).
<b>FREEDRIVE3</b>	Refers to the third available drive (for example <i>X:</i> ).
<b>FREEDRIVE4</b>	Refers to the fourth available drive (for example <i>W:</i> ).
<b>FREEDRIVE5</b>	Refers to the fifth available drive (for example <i>V:</i> ).
<b>RSPFILE</b>	Refers to the fully qualified path name of the response file specified in the <b>Response File</b> keyword in the Install Program section of the software profile.

## **INSTALL PROGRAM Keyword**

Specifies the path name, parameters, and the working directory of the install program. Only the path name is required. The token \$(key) is allowed in path name, parameters, and working directory.

The install driver starts this program in the execution of an installation request.

The install driver reports an error when an installation in service area is requested.

### **PROGRAM NAME (optional)**

The name of the CID program to execute.

### **PARAMETERS (optional)**

The parameters supplied to the program specified in **program name**.

### **WORKING DIRECTORY (optional)**

The name of the directory that the program specified in **PROGRAM NAME** is to be run in. The current directory is changed into the working directory before running the program. The working directory specified must already exist.

### **RESPONSE FILE (optional)**

Specifies the source name of the response file that is used as the input for the installation. This response file is packaged with the software object.

You must specify the special keyword \$(RSPFILE) to address this response file for the installation program.

An example is:

```
INSTALL PROGRAM:
PROGRAM NAME: $(FREEDRIVE1)\prg\install.exe
PARAMETERS: /S:$(FREEDRIVE1)\img /T:C:
```

## **BACKUP PROGRAM Keyword**

Specifies the path name, parameters, and the working directory of the backup program. Only the path name is required. The token \$(key) are allowed in path name, parameters, and working directory.

The install driver starts this program in the execution of a removable install request.

The install driver reports an error when the backup program is not specified and a removable install of the change file is requested

If specified, the remove program and the accept program must also be specified.

An example is:

```
BACKUP PROGRAM:
PROGRAM NAME: $(FREEDRIVE1)\prg\backup.exe
PARAMETERS: /T:D:
```

### REMOVE PROGRAM Keyword

Specifies the path name, parameters, and the working directory of the remove program. Only the path name is required. Tokens \$(key) are allowed in path name, parameters, and working directory.

The remove driver starts this program in the execution of a remove request or in execution of a removable install request when one installation program of the install set fails.

The remove driver reports an error in the following conditions:

- o When the remove program is not specified and the removal of the change file is requested.
- o When a remove in service area is requested.

If specified, the backup program and the accept program must be specified.

An example is:

```
REMOVE PROGRAM:  
PROGRAM NAME: $(FREEDRIVE1)\prg\remove.exe  
PARAMETERS: /T:D:
```

### ACCEPT PROGRAM Keyword

Specifies the path name, parameters, and the working directory of the accept program. Only the path name is required. Tokens \$(key) are allowed in path name, parameters, and working directory.

The accept driver starts this program in the execution of an accept request or in execution of a removable install request with autoacceptance = yes.

When the accept program is not specified, the accept driver reports an error when the acceptance of the change file is requested.

If specified, the backup program and the remove program must also be specified.

An example is:

```
ACCEPT PROGRAM:  
PROGRAM NAME: $(FREEDRIVE1)\prg\accept.exe  
PARAMETERS: /T:D:
```

### UNINSTALL PROGRAM Keyword

Specifies the path name, parameters, and the working directory of the uninstall program. Only the path name is required. Tokens \$(key) are allowed in path name, parameters, and working directory.

The uninstall driver starts this program in the execution of an uninstall request.

The uninstall driver reports an error in the following conditions:

- o When the uninstall program is not specified and the uninstallation of the change file is requested.
- o When an uninstallation in service area is requested.

An example is:

```
ACCEPT PROGRAM:
PROGRAM NAME: ${FREEDRIVE1}\prg\uninst.exe
PARAMETERS: /T:D:
```

## MAINTENANCE SYSTEM Keyword

This keyword is a special keyword to be used only when you are installing a new base operating system.

The products whose installation programs are listed under this section, are considered corequisites.

All installation programs listed under this section should be related to installing a maintenance system. Once all the products under the maintenance system are installed successfully, the main product, whose installation programs are listed before the maintenance system, is installed. This product is the base operating system.

## Structure of a CID Software Profile

To build a CID software object, you require a new profile. This profile has all the keywords defined in the software profile structure, plus some additional keywords that allow you to specify names and parameters of the Install, Backup, Remove, Accept, and Uninstall programs.

The following figure shows the structure of a CID software profile.

```

GLOBAL NAME:          Global name of change file
DESCRIPTION:         Text description of change file
LOCAL NAME:          Local name of change file on the server
CHANGE FILE TYPE:    OS2CID

PREREQ COMMAND:      Prereq command program name
POSTREQ COMMAND:     Postreq command program name

SOFTWARE PREREQUISITE:
  GLOBAL NAME        Global name of software prerequisite
  GLOBAL NAME        ...

HARDWARE PREREQUISITE:
  KEY < VALUE        Hardware prerequisite definition
  KEY > VALUE        ...

DEFAULT TOKEN:
  KEY = VALUE        Default token definition
```

## Creating CID Software Profiles

KEY = VALUE ...

```
PRE-INSTALL:          Pre-install program name
POST-INSTALL:         Post-install program name
PRE-REMOVE:          Pre-remove program name
POST-REMOVE:         Post-remove program name
PRE-ACCEPT:          Pre-accept program name
POST-ACCEPT:         Post-accept program name
PRE-UNINSTALL:       Pre-uninstall program name
POST-UNINSTALL:      Post-uninstall program name

INSTALL PROGRAM:     C/I/D install section
PROGRAM NAME:        Path name of installation program
PARAMETERS:          Parameters of installation program
WORKING DIRECTORY:   Working directory of installation program
RESPONSE FILE:       Source name of response file

BACKUP PROGRAM:      C/I/D backup section
PROGRAM NAME:        Path name of installation program
PARAMETERS:          Parameters of installation program
WORKING DIRECTORY:   Working directory of installation program

REMOVE PROGRAM:      C/I/D remove section
PROGRAM NAME:        Path name of installation program
PARAMETERS:          Parameters of installation program
WORKING DIRECTORY:   Working directory of installation program

ACCEPT PROGRAM:      C/I/D accept section
PROGRAM NAME:        Path name of installation program
PARAMETERS:          Parameters of installation program
WORKING DIRECTORY:   Working directory of installation program

UNINSTALL PROGRAM:   C/I/D uninstall section
PROGRAM NAME:        Path name of installation program
PARAMETERS:          Parameters of installation program
WORKING DIRECTORY:   Working directory of installation program

MAINTENANCE SYSTEM:
INSTALL PROGRAM:     C/I/D Install for maintenance
PROGRAM NAME:        Path name of installation program
PARAMETERS:          Parameters of installation program
WORKING DIRECTORY:   Working directory of installation program
RESPONSE FILE:       Source name of response file

INSTALL PROGRAM:     C/I/D Install for maintenance
PROGRAM NAME:        Path name of installation program
PARAMETERS:          Parameters of installation program
WORKING DIRECTORY:   Working directory of installation program
RESPONSE FILE:       Source name of response file
```

To build a CID software object, you require a new profile. This profile has all the keywords defined in the software profile structure, plus some additional keywords that allow you to specify names and parameters of the Install, Backup, Remove, Accept, and Uninstall programs.

The following figure shows the structure of a CID software profile.

```
GLOBAL NAME:          Global name of change file
DESCRIPTION:          Text description of change file
LOCAL NAME:           Local name of change file on the server
```

CHANGE FILE TYPE:	OS2CID
PREREQ COMMAND:	Prereq command program name
POSTREQ COMMAND:	Postreq command program name
SOFTWARE PREREQUISITE:	
GLOBAL NAME	Global name of software prerequisite
GLOBAL NAME	...
HARDWARE PREREQUISITE:	
KEY < VALUE	Hardware prerequisite definition
KEY > VALUE	...
DEFAULT TOKEN:	
KEY = VALUE	Default token definition
KEY = VALUE	...
PRE-INSTALL:	Pre-install program name
POST-INSTALL:	Post-install program name
PRE-REMOVE:	Pre-remove program name
POST-REMOVE:	Post-remove program name
PRE-ACCEPT:	Pre-accept program name
POST-ACCEPT:	Post-accept program name
PRE-UNINSTALL:	Pre-uninstall program name
POST-UNINSTALL:	Post-uninstall program name
INSTALL PROGRAM:	C/I/D install section
PROGRAM NAME:	Path name of installation program
PARAMETERS:	Parameters of installation program
WORKING DIRECTORY:	Working directory of installation program
RESPONSE FILE:	Source name of response file
BACKUP PROGRAM:	C/I/D backup section
PROGRAM NAME:	Path name of installation program
PARAMETERS:	Parameters of installation program
WORKING DIRECTORY:	Working directory of installation program
REMOVE PROGRAM:	C/I/D remove section
PROGRAM NAME:	Path name of installation program
PARAMETERS:	Parameters of installation program
WORKING DIRECTORY:	Working directory of installation program
ACCEPT PROGRAM:	C/I/D accept section
PROGRAM NAME:	Path name of installation program
PARAMETERS:	Parameters of installation program
WORKING DIRECTORY:	Working directory of installation program
UNINSTALL PROGRAM:	C/I/D uninstall section
PROGRAM NAME:	Path name of installation program
PARAMETERS:	Parameters of installation program
WORKING DIRECTORY:	Working directory of installation program

## Creating CID Software Profiles

```
MAINTENANCE SYSTEM:
  INSTALL PROGRAM:      C/I/D Install for maintenance
  PROGRAM NAME:        Path name of installation program
  PARAMETERS:          Parameters of installation program
  WORKING DIRECTORY:   Working directory of installation program
  RESPONSE FILE:       Source name of response file
```

```
INSTALL PROGRAM:      C/I/D Install for maintenance
  PROGRAM NAME:        Path name of installation program
  PARAMETERS:          Parameters of installation program
  WORKING DIRECTORY:   Working directory of installation program
  RESPONSE FILE:       Source name of response file
```

### Example of a CID Software Profile

Below is an example of a CID software profile used to install the OS/2 Base Version 3.0.

```
GLOBAL NAME:          OS2.BASE.REF.3.0
DESCRIPTION:          OS/2 Base Version 3.0
LOCAL NAME:           $(REPOSITORY)\os2base.cf

CHANGE FILE TYPE:    OS2CID
PREREQ COMMAND:      mount -u0 -g0 $(FREEDRIVE1) z: ▶
                    p57pt02:c:\images
POSTREQ COMMAND:     umount $(FREEDRIVE1)

INSTALL PROGRAM:
  PROGRAM NAME:       $(FREEDRIVE1)\prg\seinst.exe
  PARAMETERS:        /S:$(FREEDRIVE1)\img ▶
                    /T:C:

BACKUP PROGRAM:
  PROGRAM NAME:       $(FREEDRIVE1)\prg\seback.exe
  PARAMETERS:        /T:D:

REMOVE PROGRAM:
  PROGRAM NAME:       $(FREEDRIVE1)\prg\seremv.exe
  PARAMETERS:        /T:D:

ACCEPT PROGRAM:
  PROGRAM NAME:       $(FREEDRIVE1)\prg\seaccp.exe
  PARAMETERS:        /T:D:
```

## Index

### Special Characters

- o, connect command 53
- r, connect command 52
- r, disconnect command 71
- s, connect command 52
- t, telephone number 52

### A

- acc command 20
- access keys
- act command 25
- active, change file status 116
- addgp command 29
- addpm command 31
- addprf command 33
- addtg command 36
- addusr command 41
- auth command 43
- authorizations
  - addusr command 41
  - auth command 43
  - command-line interface 16
  - lsusr command 133
  - unauth command 191
- available, change file status 115

### B

- back level, change file status 115
- bld command 47

### C

- cat command 48
- catalogs
  - cat command 48
  - ls command 107
  - lsak command 109
  - uncat command 198
- change file 115
- change file profiles
  - creating 249
  - dynamic 249
  - envelope keywords 252

- change file profiles (*continued*)
  - examples 266
  - examples of 266
  - keywords 252
  - static 249
  - structure 249
  - syntax 249
- change files
  - acc command 20
  - active status 116
  - available, status 115
  - back level, status 115
  - bld command 47
  - building change files 47
  - creating profiles 249
  - del command 54
  - delcm command 58
  - discovered, status 115
  - distributed, status 115
  - distribution pending, status 115
  - example of profile 266
  - imp command 91
  - in error status 115
  - inactive status 116
  - inst command 92
  - installed status 115
  - lscf command 111
  - lscm command 114
  - not available status 115
  - not removable status 116
  - reboot required, status 116
  - rem command 148
  - removable status 116
  - rtv command 158
  - scheduled, status 115
  - send command 168
  - unbld command 195
  - unpacking change files 195
  - updcn command 207
  - vercm command 229
- change management
  - acc command 20
  - act command 25
  - delprf command 66
  - delrq command 67
  - lsrq command 124

change management (*continued*)

- rem command 148
- uninst command 199
- vercm command 229
- CID software objects 268
- CID software profile 275
  - software profile example 278

clients

command-line interface 1

- acc 20
- accepting change files 20
- act 25
- activating changes on targets 25
- addgp 29
- adding parameters to targets 31
- adding target groups 29
- adding user profiles 33
- adding users 41
- addpm 31
- addprf 33
- addtg 36
- addusr 41
- authorizations 16
- authorizing targets to access files 43
- bld 47
- cat 48
- checking communication status 179
- closing command line session 141
- connect 52
- connect to server from mobile client 52
- copying files from servers 82
- creating catalog entries 48
- creating targets 36
- dates 12
- del 54
- delcm 58
- delete local files 54
- deleting authorization profiles 66
- deleting change file status 58
- deleting parameters from targets 65
- deleting requests 67
- deleting target groups 63
- deleting targets 69
- deleting users 70
- delgp 63
- delpm 65
- delprf 66
- delrq 67
- deltg 69
- delusr 70

command-line interface (*continued*)

- disconnect 71
- disconnect mobile client connection 71
- displaying message log 104
- displaying queue entry details 121
- entering data 10
- eraserq 72
- erasing requests 72
- examples of date and time usage 14
- exec 74
- execf 78
- executing noncataloged programs or scripts 78
- executing programs or scripts 74
- exp 82
- giving priority to a queue 139
- hdlrq 88
- help 84
- hldc 85
- hldq 86
- hldtg 90
- holding queues 86
- holding remote communications 85
- holding requests 88
- holding targets 90
- imp 91
- importing change files 91
- inst 92
- installing change files 92
- inv 99
- listing access keys 109
- listing authorization profiles 119
- listing catalog entries 107
- listing change file contents 111
- listing request status 124
- listing target groups 117
- listing target status 182
- listing targets 129
- listing user authorizations 133
- log 104
- ls 107
- lsak 109
- lsbs 110
- lscf 111
- lscm 114
- lsgp 117
- lsprf 119
- lsq 121
- lsrq 124
- lstg 129
- lsusr 133



- delpm command 65
- delprf command 66
- delrq command 67, 70
- deltg command 69
- disconnect command 71
- discovered, change file status 115
- diskcamera command line operations 233
- diskcamera error codes 237
- diskcamera modification file 239
- distributed, change file status 115
- distribution
  - lsq command 121
  - lsrq command 124
  - prgq command 137
  - rtrv command 158
  - rtrvf command 163
  - send command 168
  - sendf command 173
- distribution pending, change file status 115
- duration, connection window 52
- dynamic change file 249
- dynamic section, change file profile 265

## E

- envelope keywords, change file profile 252
- eraserq command 72
- Error codes for diskcamera 237
- exec command 74
- execf command 78
- execpln command 88
- exp command 82

## G

- global name 12

## H

- help command 84
- hldc command 85
- hldq command 86
- hldtg command 90
- host deletion of pending requests xiii

## I

- imp command 91
- in error, change file status 115

- inactive, change file status 116
- inst command 92
- installed, change file status 115
- installp change files
  - change file profile keywords 252
  - creating profiles 249
- inv command 99
- inventory discovery 99

## L

- local target statuses, changes to xiv
- log command 104
- ls command 107
- lsak command 109
- lsbs command 110
- lscf command 111
- lscm command 114
- lsgp command 117
- lsprf command 119
- lsq command 121
- lsrq command 124
- lstg command 129
- lsusr command 133

## M

- messages
  - command-line interface 15
  - log command 104
  - troff command 189
  - tron command 190
- modification file 239
- modification file, rules for 246

## N

- new in this release xi
- not available, change file status 115
- not removable, change file status 116
- nvdn command 135

## O

- open command 136
- outstanding request, connection window 53

## P

- pending requests, delete xiii
- platforms supported xi
- prgq command 137
- pristine installation scenarios, new xi
- profiles, change file 249
  - creating 249
  - examples 266
  - keywords 252
- prtyq command 139

## Q

- queues
  - delrq command 67
  - eraserq command 72
  - hldc command 85
  - hldq command 86
  - lsq command 121
  - lsrq command 124
  - prgq command 137
  - prtyq command 139
  - relc command 142
  - relq command 143
- quit command 141

## R

- reboot required, change file status 116
- recursion, connection window 52, 71
- recursive requests
- relc command 142
- relq command 143
- relrq command 145
- reltg command 147
- rem command 148
- removable, change file status 116
- rentg command 153
- reset command 155
- return codes, command-line interface 15
- rld command 156
- rstrq command 157
- rtrv command 158
- rules for the modification file statements 246

## S

- scheduled, change file status 115

## scripts

- exec command 74
- execf command 78
- send command 163, 168, 173
- server
  - exp command 82
  - svr command 186
- start command 178
- start time, connection window 52
- stat command 179
- stattg command 182
- stattg command, changes to statuses xiv
- status 115
  - active, change file 116
  - available, change file 115
  - back level, change file 115
  - delcm command 58
  - discovered, change file 115
  - distributed, change file 115
  - distribution pending, change file 115
  - in error, change file 115
  - inactive, change file 116
  - installed, change file 115
  - lscm command 114
  - not available, change file 115
  - not removable, change file 116
  - reboot required, change file 116
  - removable, change file 116
  - scheduled, change file 115
  - stat command 179
- stop command 185
- svr command 186
- syntax rules, command-line interface 8

## T

- target name, connect command 52
- targets
  - act command 25
  - addgp command 29
  - addpm command 31
  - addtg command 36
  - auth command 43
  - delgp command 63
  - delpm command 65
  - deltg command 69
  - entering names, command-line interface 11
  - exec command 74
  - execf command 78
  - hldtg command 90

targets (*continued*)  
  lsak command 109  
  lsgp command 117  
  lstg command 129  
  reltg command 147  
  rem command 148  
  rentg command 153  
  stattg command 182  
  unauth command 191  
  updtg command 220  
telephone number, connect command 52  
time, command-line interface 12  
TME 10 Software Distribution  
  reset command 155  
  start command 178  
  stop command 185  
trace command 187  
transmission plans  
  deleting requests 67  
  eraserq command 72  
  hldrq command 88  
  lsrq command 124  
  relrq command 145  
  rstrq command 157  
  updrq command 216  
troff command 189  
tron command 190

## U

unauth command 191  
unbld command 195  
uncat command 198  
uninst command 199  
updak command 205  
updbd command 204, 206  
updcn command 207  
updprf command 212  
updpwd command 215  
updrq command 216  
updtg command 220  
updusr command 227  
users  
  addprf command 33  
  addusr command 41  
  delprf command 66  
  delusr command 70  
  lsprf command 119  
  lsusr command 133  
  updak command 205

users (*continued*)  
  updprf command 212  
  updpwd command 215  
  updusr command 227

## V

vercm command 229

## W

what's new xi



---

# Communicating Your Comments to IBM

TME® 10 Software Distribution  
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  
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.

<b>Overall, how satisfied are you with the information in this book?</b>	Satisfied	Dissatisfied
	<input type="checkbox"/>	<input type="checkbox"/>

<b>How satisfied are you that the information in this book is:</b>	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.

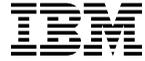
Name \_\_\_\_\_ Address \_\_\_\_\_

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