

AIX 5L バージョン 5.3



コマンド・リファレンス 第 4 巻 (n から r)



AIX 5L バージョン 5.3



コマンド・リファレンス 第 4 巻 (n から r)

お願い

本書および本書で紹介する製品をご使用になる前に、929 ページの『特記事項』に記載されている情報をお読みください。

本書は AIX 5L バージョン 5.3 および新しい版で明記されていない限り、このプロダクトの以降のすべてのリリースに適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： SC23-4891-07  
AIX 5L Version 5.3  
Commands Reference, Volume 4, n - r

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第8刷第1刷 2010.4

© Copyright International Business Machines Corporation 1997, 2010.

---

# 目次

本書について	xi
本書の使用方法	xi
強調表示	xi
ISO 9000	xiv
Single UNIX Specification の 32 ビットおよび 64 ビット・サポート	xiv
関連情報	xiv
コマンドのアルファベット順リスト	1
named デーモン	1
named8 デーモン	2
named9 デーモン	5
namerslv コマンド	7
ncheck コマンド	10
nddctl コマンド	11
ndp コマンド	12
ndpd-host デーモン	13
ndpd-router デーモン	14
ndx コマンド	19
neqn コマンド	21
netpmon コマンド	22
netstat コマンド	33
newaliases コマンド	45
newform コマンド	45
newgrp コマンド	48
newkey コマンド	49
news コマンド	51
next コマンド	52
nfs.clean コマンド	54
nfs4cl コマンド	55
nfs4smctl コマンド	57
nfsd デーモン	58
nfshostkey コマンド	60
nfshostmap コマンド	61
nfso コマンド	62
nfsrgyd デーモン	84
nfsstat コマンド	85
nice コマンド	89
nim コマンド	91
nim_clients_setup コマンド	104
nim_master_recover コマンド	106
nim_master_setup コマンド	109
nim_move_up コマンド	112
nim_update_all コマンド	122
nimadapters コマンド	124
nimadm コマンド	130
nimclient コマンド	139
nimconfig コマンド	142
nimdef コマンド	145
niminit コマンド	150

niminv コマンド	153
nimol_backup コマンド	160
nimol_config コマンド	162
nimol_install コマンド	165
nimol_lslpp コマンド	167
nimol_update コマンド	169
nimquery コマンド	170
nis_cachemgr デーモン	172
nisaddcred コマンド	173
nisaddent コマンド	177
niscat コマンド	181
nischgrp コマンド	182
nischmod コマンド	184
nischown コマンド	186
nischttl コマンド	187
nisclient コマンド	189
nisdefaults コマンド	192
niserror コマンド	195
nisgrep コマンド	195
nisgrpadm コマンド	197
nisinit コマンド	198
nisln コマンド	201
nislog コマンド	201
nisls コマンド	202
nismatch コマンド	204
nismkdir コマンド	205
nismkuser コマンド	207
nisping コマンド	209
nispopulate コマンド	211
nisrm コマンド	215
nisrmdir コマンド	215
nisrmuser コマンド	217
nisserver コマンド	218
nissetup コマンド	220
nisshowcache コマンド	221
nisstat コマンド	222
nistbladm コマンド	223
nistest コマンド	227
nistoldif コマンド	229
nisupdkeys コマンド	231
nl コマンド	233
nlsrc コマンド	235
nm コマンド	238
nmon コマンド	241
no コマンド	260
nohup コマンド	311
notifyevent コマンド	313
nrglbd デーモン	314
nroff コマンド	315
nslookup コマンド	318
nsupdate コマンド	324
nsupdate4 コマンド	325

nsupdate8 コマンド	327
nsupdate9 コマンド	330
ntpdate コマンド	334
ntpq コマンド	336
ntptrace コマンド	341
nulladm コマンド	343
number コマンド	343
od コマンド	344
odmadd コマンド	349
odmchange コマンド	350
odmcreate コマンド	351
odmdelete コマンド	354
odmdrop コマンド	355
odmget コマンド	356
odmshow コマンド	357
on コマンド	357
OS_install コマンド	359
oslevel コマンド	363
ospf_monitor コマンド	364
pac コマンド	367
pack コマンド	369
packf コマンド	371
pagdel コマンド	373
pagesize コマンド	374
paginit コマンド	375
paglist コマンド	376
panel20 コマンド	376
passwd コマンド	377
paste コマンド	380
patch コマンド	383
pathchk コマンド	388
pax コマンド	389
pcat コマンド	404
pdelay コマンド	405
pdisable コマンド	406
penable コマンド	407
perfwb コマンド	408
pg コマンド	409
phold コマンド	412
pic コマンド	413
pick コマンド	420
ping コマンド	424
pioattred コマンド	428
piobe コマンド	430
pioburst コマンド	432
piocnvt コマンド	434
piodigest コマンド	435
piodmgr コマンド	437
piofontin コマンド	438
pioformat コマンド	439
piofquote コマンド	441
piolsvp コマンド	442

piomgpdev コマンド	445
piomkapqd コマンド	446
piomkpg コマンド	448
piomsg コマンド	450
pioout コマンド	452
pioptdef コマンド	454
pkgadd コマンド	456
pkgask コマンド	459
pkgchk コマンド	461
pkginfo コマンド	463
pkgmk コマンド	465
pkgparam コマンド	467
pkgproto コマンド	469
pkgrm コマンド	471
pkgtrans コマンド	472
platform_dump コマンド	474
plotgbe コマンド	476
plotlbe コマンド	478
pmcycles コマンド	479
pmlist コマンド	480
pmtu コマンド	482
pop3d デーモン	483
pop3ds デーモン	485
portmap デーモン	486
portmir コマンド	487
post コマンド	489
pppattachd デーモン	491
pppcontrold デーモン	493
pppdial コマンド	499
pppstat コマンド	501
pprof コマンド	503
pr コマンド	504
prctmp コマンド	507
prdaily コマンド	508
preparevsd コマンド	509
preprnode コマンド	511
prev コマンド	513
printenv コマンド	515
printf コマンド	515
proccred コマンド	519
procfiles コマンド	520
procflags コマンド	522
procldd コマンド	523
procmap コマンド	524
procrun コマンド	526
procsig コマンド	527
procstack コマンド	528
procstop コマンド	530
proctree コマンド	531
procwait コマンド	532
procwdx コマンド	533
prof コマンド	534



proff コマンド	537
projctl コマンド	538
prompter コマンド	547
proto コマンド	549
proxymngr コマンド	550
prs コマンド (SCCS)	552
prtacct コマンド	556
prtconf コマンド	557
ps コマンド	561
ps4014 コマンド	583
ps630 コマンド	584
psc または psdit コマンド	585
pshare コマンド	588
psplot コマンド	589
psrev コマンド	590
psroff コマンド	591
pstart コマンド	594
pstat コマンド	595
ptx コマンド	596
pwchange コマンド	598
pwck コマンド	600
pwd コマンド	601
pwdadm コマンド	602
pwdck コマンド	604
pwtokey コマンド	607
pxed コマンド	610
qadm コマンド	611
qcan コマンド	613
qchk コマンド	614
qdaemon コマンド	616
qhld コマンド	618
qmov コマンド	619
qosadd コマンド	620
qoslist コマンド	621
qosmod コマンド	622
qosremove コマンド	624
qosstat コマンド	625
qpri コマンド	627
qprt コマンド	628
qstatus コマンド	637
quiz コマンド	639
quot コマンド	641
quota コマンド	643
quotacheck コマンド	644
quotaon または quotaoff コマンド	646
raddbm コマンド	647
ranlib コマンド	651
raso コマンド	652
ras_logger コマンド	660
rc コマンド	662
rc.mobip6 コマンド	663
rc.powerfail コマンド	664

rcp コマンド	666
rcvdist コマンド	671
rcvpack コマンド	672
rcvstore コマンド	672
rcvtty コマンド	674
rdist コマンド	675
rdistd コマンド	689
rdump コマンド	689
read コマンド	692
readlvcopy コマンド	694
reboot または fastboot コマンド	695
recreatevg コマンド	696
recsh コマンド	698
redefinevg コマンド	699
reducevg コマンド	700
refer コマンド	702
refile コマンド	704
refresh コマンド	707
refsrc コマンド	708
refsensor コマンド	710
regcmp コマンド	713
rembak コマンド	714
remove コマンド	716
removevsd コマンド	717
renice コマンド	718
reorgvg コマンド	720
repl コマンド	722
replacepv コマンド	725
repquota コマンド	727
reset コマンド	728
resize コマンド	730
restart-secdapclntd コマンド	731
restbase コマンド	732
restore コマンド	733
restorevgfiles コマンド	744
restvg コマンド	746
resumevsd コマンド	749
rev コマンド	750
revnetgroup コマンド	751
rexd デーモン	752
rexec コマンド	753
rexcld デーモン	754
rgb コマンド	756
ripquery コマンド	757
rksh コマンド	758
rlogin コマンド	761
rlogind デーモン	764
rm コマンド	766
rm_niscachemgr コマンド	769
rm_nisd デーモン	771
rm_nispasswdd デーモン	772
rmail コマンド	773

rmadrec コマンド	774
rmC2admin コマンド	778
rmCCadmin コマンド	779
rmcctrl コマンド	780
rmcifscred コマンド	783
rmcifsmnt コマンド	784
rmclass コマンド	785
rmcomg コマンド	787
rmcondition コマンド	789
rmcondresp コマンド	791
rmcosi コマンド	795
rmdel コマンド	796
rmdev コマンド	797
rmdir コマンド	799
rmf コマンド	801
rmfilt コマンド	802
rmfs コマンド	803
rmgroup コマンド	804
rmiscsi コマンド	806
rmitab コマンド	808
rmkeyserv コマンド	808
rmlpcmd コマンド	809
rmlv コマンド	812
rmlvcopy コマンド	814
rmm コマンド	815
rmnamsv コマンド	817
rmnfs コマンド	817
rmnfsexp コマンド	818
rmnfsmnt コマンド	819
rmnfsproxy コマンド	820
rmnotify コマンド	821
rmpath コマンド	822
rmprtsv コマンド	824
rmprsv コマンド	825
rmqos コマンド	827
rmque コマンド	828
rmquedev コマンド	829
rmramdisk コマンド	830
rmresponse コマンド	831
rmrole コマンド	834
rmrpdomain コマンド	835
rmrpnod コマンド	837
rmrset コマンド	840
rmrsrc コマンド	841
rmsensor コマンド	845
rmserver コマンド	847
rmsock コマンド	848
rmss コマンド	849
rmssys コマンド	853
rmt コマンド	855
rmtcpip コマンド	856
rmts コマンド	857

rmtun コマンド	858
rmuser コマンド	859
rmvfs コマンド	861
rmvirprt コマンド	861
rmyp コマンド	863
rndc コマンド	864
rndc-confgen コマンド	865
roffbib コマンド	866
rollback コマンド	867
route コマンド	868
routed デーモン	872
rpc.nisd デーモン	877
rpc.nispasswd デーモン	879
rpc.pcnfsd デーモン	880
rpcgen コマンド	882
rpcinfo コマンド	884
rrestore コマンド	887
Rsh コマンド	890
rsh または remsh コマンド	892
rshd デーモン	896
rstatd デーモン	899
rtl_enable コマンド	900
runacct コマンド	902
runact コマンド	905
runcat コマンド	909
runlpcmd コマンド	910
rup コマンド	914
ruptime コマンド	915
ruser コマンド	916
rusers コマンド	918
rusersd デーモン	920
rvsdrestrict コマンド	920
rwall コマンド	922
rwalld デーモン	923
rwho コマンド	924
rwhod デーモン	925
<b>付録. 特記事項</b>	<b>929</b>
商標	930
<b>索引</b>	<b>931</b>

---

## 本書について

本書は、AIX オペレーティング・システムのコマンドについての詳細情報を網羅して、エンド・ユーザーの皆様にご提供するものです。コマンドはアルファベット順にカテゴリー別にリストされ、コマンドおよびその使用可能なフラグに関する完全な説明が記載されています。利用可能な場合、各コマンドのリスト表示には例が含まれます。本書 (第 4 巻) には、先頭文字が n から r までの AIX コマンドが記載してあります。なお、本資料は、オペレーティング・システムに付属して配送されている文書 CD にも収められています。

---

## 本書の使用方法

コマンドとは、操作やプログラムの実行の要求です。ユーザーは、オペレーティング・システムに実行させたいタスクを指示する際に、コマンドを使います。コマンドが入力されると、コマンド・インタープリター (シェルとも呼ばれます) によって解釈され、そのタスクが処理されます。

一部のコマンドは 1 ワードをタイプするだけで入力できます。また、複数のコマンドを組み合わせ、あるコマンドからの出力を別のコマンドの入力とすることができます。これをパイプライン処理と呼びます。

フラグは、コマンドのアクションをさらに詳しく定義します。フラグとは、コマンド行上でコマンド名と一緒に使用される修飾子で、通常は前にダッシュが付きます。

また、コマンドはグループにまとめてファイルに格納することもできます。このようなコマンドをシェル・プロシージャまたはシェル・スクリプトと呼びます。ユーザーはコマンドを個別に実行する代わりに、そのコマンドが入っているファイルを実行します。

一部のコマンドは、Web-based System Manager アプリケーション、または System Management Interface Tool (SMIT) を使用して作成できます。

---

## 強調表示

本書では、以下の強調表示規則を使用します。

太字	コマンド、サブルーチン、キーワード、ファイル、構造体、ディレクトリー、およびシステムによって名前が事前に定義されているその他の項目を表します。また、ユーザーが選択する視覚的なオブジェクト (ボタン、ラベル、アイコンなど) も示します。
イタリック	実際の名前または値をユーザーが指定する必要のあるパラメーターを示します。
Monospace (モノスペース)	特定のデータの値の例、実際に表示されるものと同様のテキストの例、プログラマーが作成するものと同様のプログラム・コード部分の例、システムからのメッセージ、または実際に入力する情報を示します。

## フォーマット

各コマンドの説明は、以下のセクションのいくつかからなります。

目的	各コマンドの主な機能の説明。
構文	コマンド行オプションを示す構文ステートメント。
説明	機能と使用方法を詳細に説明したコマンド解説。

フラグ	コマンド行フラグと関連パラメーターのリスト、およびフラグがコマンドのアクションを変更する方法に関する説明。
パラメーター	コマンド行パラメーターのリストとその説明。
サブコマンド	サブコマンド (対話式コマンドの場合) の使用方法を説明するリスト。
終了状況	コマンドが戻す終了値の説明。
セキュリティ	コマンドを実行するために必要な許可の指定。
例	コマンドの使用法を示す特定の例。
ファイル	コマンドで使用するファイルのリスト。
関連情報	本書の関連コマンドと他のマニュアル内の関連説明のリスト。

## 構文ステートメントの読み方

構文ステートメントは、コマンド構文を表現しており、大括弧 ([ ])、中括弧 ({ }), 垂直線 (|) などの記号が使用されています。以下に **unget** コマンドの構文ステートメントの例を示します。

```
unget [ -rSID ] [ -s ] [ -n ] File ...
```

構文ステートメントには、次の規則があります。

- **太字**の項目は、示された文字どおりにコマンド行に入力します。太字の項目にはコマンド名、フラグ、およびリテラル文字があります。
- *イタリック体*の項目は、実際の名前で置換しなければならない変数を表します。このような項目には、*Files* や *Directories* など、フラグのあとに続くパラメーターや、コマンドが読み取るパラメーターがあります。
- 大括弧で囲まれたパラメーターはオプションです。
- 中括弧で囲まれたパラメーターは必須です。
- 大括弧にも中括弧にも囲まれていないパラメーターは、必須です。
- 垂直線は、パラメーターを 1 つだけ選択することを示します。例えば [ a | b ] は、a を選択する、b を選択する、または何も選択しないことができることを示します。同様に、{ a | b } は、a または b のどちらかを選択しなければならないことを示します。
- 省略符号 ( ... ) は、コマンド行にパラメーターを繰り返し指定できることを示します。
- ダッシュ ( - ) は標準入力を表します。

## インストール可能なソフトウェア・パッケージのリスト表示

個々のコマンドから成るインストール可能なソフトウェア・パッケージ (ファイルセット) をリストするには、**-w** フラグが付いた **lspp** コマンドを使用します。例えば、**installp** コマンドを含むファイルセットをリストするには、次のように入力します。

```
lspp -w /usr/sbin/installp
```

以下のような結果が表示されます。

File	Fileset	Type
-----	-----	-----
/usr/sbin/installp	bos.rte.install	File

**installp** という語が含まれるすべてのファイル名を持つファイルセットをリストするには、次のように入力します。

```
lspp -w "*installp*"
```

以下のような結果が表示されます。

File	Fileset	Type
/usr/sbin/installp	bos.rte.install	File
/usr/clvm/sbin/linstallpv	prpq.clvm	File
/usr/lpp/bos.sysmgmt/nim/methods/c_installp	bos.sysmgmt.nim.client	File

## バックグラウンドでのコマンドの実行

処理に長時間かかるコマンドを実行したいときは、コマンドをバックグラウンドで実行するように指定できます。バックグラウンド処理は、処理が低速のプログラムを実行するときに有効な方法です。コマンドをバックグラウンドで実行するには、次のようにコマンドの最後に `&` 演算子を使用します。

*Command&*

処理がバックグラウンドで実行されている間に、システム上で作業を続行して別のコマンドを入力できます。

また、指定した時刻または特定の日に、コマンドを実行できます。**cron** デーモンを使用すると、自動実行するコマンドをスケジュールすることができます。**at** コマンドおよび **batch** コマンドを使用すると、コマンドをあとで実行したり、システム負荷のレベルに余裕があるときに実行できます。

## コマンドの入力

一般に、コマンド行上のシェル・プロンプトの後にコマンドを入力します。シェル・プロンプトは変わる場合があります。次の例では、`$` がプロンプトです。

例えば、現行ディレクトリーの内容のリストを表示するには、次のように `ls` とタイプして `Enter` キーを押します。

```
$ ls
```

コマンドを入力して実行している間、オペレーティング・システムからはシェル・プロンプトは表示されません。コマンドがアクションを完了すると、システムは再びプロンプトを表示します。これは、別のコマンドを入力できることを示します。

コマンドの一般的な入力フォーマットは次のとおりです。

*Command Flag(s) Parameter*

フラグは、コマンドの機能を変更します。多くのコマンドには複数のフラグがあります。例えば、**ls** コマンドの後に **-l** (long) フラグをタイプすると、システムは現行ディレクトリーの内容に関する詳細情報を表示します。下記の例は、**ls** コマンドと **-l** フラグの使用方法を示しています。

```
$ ls -l
```

パラメーターは、コマンドまたはフラグの後に続く文字列からなっています。パラメーターは、ファイル名、ディレクトリー名、値などのデータを指定します。下記の例では、**/usr/bin** という名前のディレクトリーがパラメーターです。

```
$ ls -l /usr/bin
```

コマンドを入力するときは、必ず以下の点に注意してください。

- コマンドは通常、小文字で入力します。
- フラグの前には `-` (負符号) を付けます。
- 各コマンドを `;` (セミコロン) で区切ると、コマンド行に複数のコマンドをタイプできます。

- \ (バックスラッシュ) を使うと、長いコマンドのシーケンスを次行に続けることができます。バックスラッシュは 1 行目の最後に配置します。次の例は、バックスラッシュの配置を示しています。

```
$ cat /usr/ust/mydir/mydata > \  
/usr/usts/yourdir/yourdata
```

ある種のコマンドを入力すると、シェル・プロンプトが変化します。一部のコマンドは実際にはプログラムなので (**telnet** コマンドなど)、そのコマンドの実行中はプロンプトが変わります。プログラム内で発行するコマンドをサブコマンドと呼びます。プログラムを終了すると、プロンプトはシェル・プロンプトに戻ります。

オペレーティング・システムはさまざまなシェル (Bourne、C、または Korn など) で動作可能であり、入力されたコマンドはシェルによって解釈されます。したがって、正しいフォーマットでコマンドを入力できるように、どのシェルを使用中なのか知っておく必要があります。

## コマンドの停止

コマンドを入力してから、そのコマンドの実行を停止したい場合は、コマンドをそれ以上処理させずに中断できます。処理中のコマンドを停止するには、割り込みキー・シーケンス (通常は Ctrl-C または Alt-Pause) を押します。処理が停止されると、シェル・プロンプトが表示され、別のコマンドを入力できます。

---

## ISO 9000

当製品の開発および製造には、ISO 9000 登録品質システムが使用されました。

---

## Single UNIX Specification の 32 ビットおよび 64 ビット・サポート

バージョン 5.2 以降、AIX オペレーティング・システムは、UNIX ベースのオペレーティング・システムのポータビリティに関する「The Open Group's Single UNIX Specification Version 3 (UNIX 03)」をサポートするように設計されています。数多くの新規インターフェースが追加され、また数多くの現行インターフェースがこの仕様を満たすように拡張されました。これによりバージョン 5.2 は、AIX の以前のリリースとの互換性を保ちながら、さらにオープンになり、アプリケーションのポータビリティを高めています。

UNIX 03 ポータブル・アプリケーションを開発する正しい方法を判断するには、「The Open Group's UNIX 03 specification」を参照することが必要になる場合があります。この仕様は、<http://www.unix.org/> でオンライン・アクセスができ、またダウンロードすることができます。

---

## 関連情報

以下の資料には、コマンドについての情報や関連情報が説明されています。

- AIX 5L バージョン 5.3 コマンド・リファレンス 第 1 巻
- AIX 5L バージョン 5.3 コマンド・リファレンス 第 2 巻
- AIX 5L バージョン 5.3 コマンド・リファレンス 第 3 巻
- AIX 5L バージョン 5.3 コマンド・リファレンス 第 4 巻
- AIX 5L バージョン 5.3 コマンド・リファレンス 第 5 巻
- AIX 5L バージョン 5.3 コマンド・リファレンス 第 6 巻
- AIX 5L Version 5.3 Files Reference
- プリンターおよび印刷



- インストールおよび移行
- *AIX 5L* バージョン 5.3 パーティション環境での *AIX* インストール
- *AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*
- パフォーマンス・マネージメント
- *AIX 5L Version 5.3 Performance Tools Guide and Reference*
- セキュリティー
- オペレーティング・システムおよびデバイスのマネージ
- ネットワークおよびコミュニケーションのマネージ
- *AIX 5L Version 5.3 Technical Reference: Base Operating System and Extensions Volume 1*
- *AIX 5L Version 5.3 Technical Reference: Base Operating System and Extensions Volume 2*
- *AIX 5L Version 5.3 Technical Reference: Communications Volume 1*
- *AIX 5L Version 5.3 Technical Reference: Communications Volume 2*
- *AIX 5L Version 5.3 Technical Reference: Kernel and Subsystems Volume 1*
- *AIX 5L Version 5.3 Technical Reference: Kernel and Subsystems Volume 2*
- *AIX 5L Version 5.3 Web-based System Manager* 管理ガイド
- *Performance Toolbox Version 2 and 3 for AIX: Guide and Reference*



---

# コマンドのアルファベット順リスト

---

## named デーモン

### 目的

ドメイン名プロトコルのサーバー機能を提供します。

### 構文

**named8** または **named9** デーモンのいずれかの構文を参照してください。

### 説明

AIX は、3 つのバージョン (4、8、9) の BIND をサポートします。デフォルトでは、**named** は、**nsupdate** から **nsupdate4** まで、**named-xfer** から **named-xfer4** までにリンクします。別のバージョンの **named** を使用する場合は、これに応じて **named** および **named-xfer** デーモンのシンボリック・リンクを再リンクする必要があります。

例えば **named8** を使用するには次のようにします。

```
ln -fs /usr/sbin/named8 /usr/sbin/named
ln -fs /usr/sbin/named8-xfer /usr/sbin/named-xfer
```

**nsupdate4** は **named8** で使用できますが、**nsupdate9** はセキュリティ処理が異なるため、**named9** で使用する必要があります。**named9** を使用する場合は、このデーモンは **named-xfer** を使用しないので、どれをリンクしても問題ありません。

### ファイル

<b>/usr/sbin/named</b>	<b>named</b> デーモンが入っています。
<b>/usr/sbin/named8</b>	<b>named8</b> デーモンが入っています。
<b>/usr/sbin/named9</b>	<b>named9</b> デーモンが入っています。
<b>/etc/resolv.conf</b>	ドメイン・ネーム・サービスの使用方法を指定します。
<b>/etc/rc.tcpip</b>	システムを再始動するたびにデーモンを初期化します。
<b>/etc/named.pid</b>	プロセス ID を格納します。
<b>/etc/services</b>	ソケット・サービスの割り当てを定義します。
<b>/usr/samples/tcpip/named.boot</b>	サンプルの <b>named.boot</b> ファイルがその使い方の説明と一緒に入っています。
<b>/usr/samples/tcpip/named.data</b>	サンプルの DOMAIN データ・ファイルが、その使用上の指示と一緒に入っています。
<b>/usr/samples/tcpip/hosts.awk</b>	<b>/etc/hosts</b> ファイルを <b>/etc/named.data</b> ファイルに変換するためのサンプル <b>awk</b> スクリプトが入っています。このファイルにはその使い方の説明も入っています。
<b>/usr/samples/tcpip/addrns.awk</b>	<b>/etc/hosts</b> ファイルを <b>/etc/named.rev</b> ファイルに変換するためのサンプル <b>awk</b> スクリプトが入っています。このファイルにはその使い方の説明も入っています。
<b>/usr/samples/tcpip/named.dynamic</b>	動的データベース・セットアップが入っています。

## 関連情報

**nslookup** コマンド、**traceroute** コマンド、**kill** コマンド。

**named8** および **named9** デーモン。

**named.conf** ファイル・フォーマット、**DOMAIN Cache** ファイル・フォーマット、**DOMAIN Data** ファイル・フォーマット、**DOMAIN Reverse Data** ファイル・フォーマット、**DOMAIN Local Data** ファイル・フォーマット、**resolv.conf** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ のネーム・サーバー・レゾリューションのセクション、およびドメイン・ネーム・レゾリューションの計画のセクション。

---

## named8 デーモン

### 目的

ドメイン名プロトコルのサーバー機能を提供します。

### 構文

```
/usr/sbin/named8 [ -d DebugLevel ] [ -p PortNumber ] [ -c ConfFile ] [ -w WorkingDirectory ] [ -t RootDirectory ] [ -q ] [ -r ] [ -f ]
```

### 説明

**/usr/sbin/named8** デーモンはドメイン名プロトコル (DOMAIN) 用のサーバーです。 **named8** デーモンは、ネーム・サーバー・ホスト上で実行し、ドメイン・ネーム・レゾリューション機能を制御します。

どのネーム・サーバー・デーモンを使用するかを選択は、**/usr/sbin/named** シンボリック・リンクと **/usr/sbin/named-xfer** シンボリック・リンクによって制御されます。

**注:** **named8** デーモンは、システム・リソース・コントローラー (SRC) または System Management Interface Tool (SMIT) を使用して制御できます。システム始動時に毎回デーモンを始動するには、**rc.tcpip** ファイルを使用します。

**named8** デーモンは外部ホストで実行している resolver ルーチンによって作成されるネーム・サーバー要求を listen します。デーモンは、**/etc/services** ファイルに定義したソケットを listen します。

**/etc/services** ファイルのエントリは **domain** で始まります。ただし、このソケット割り当ては、コマンド行に **-p** フラグを使ってオーバーライドできます。

**注:** **/etc/resolv.conf** ファイルは、ローカル・カーネルと resolver ルーチンに DOMAIN プロトコルを使用するように通知します。DOMAIN ネーム・サーバー・ホスト上で **named8** デーモンを使用するには、**/etc/resolv.conf** ファイルが存在していて、ローカル・ホストのアドレスまたはループバック・アドレス (127.0.0.1) を含んでいなければなりません。**/etc/resolv.conf** ファイルが存在しない場合、ローカル・カーネルと resolver ルーチンは、**/etc/hosts** データベースを使用します。この場合、**named8** デーモンは正しく機能しません。

システム・リソース・コントローラーを使用した **named8** デーモンの操作

**named8** デーモンは、システム・リソース・コントローラー (SRC) によって制御されるサブシステムです。 **named8** デーモンは **tcpip** システム・グループのメンバーです。このデーモンはデフォルトでは使

用不可で、以下の SRC コマンドにより操作できます。

<b>startsrc</b>	サブシステム、サブシステムのグループ、あるいはサブサーバーを開始します。
<b>stopsrc</b>	サブシステム、サブシステムのグループ、あるいはサブサーバーを停止します。
<b>refresh</b>	<b>named8</b> デーモンが <b>/etc/named.conf</b> ファイルの再読み取りを実行します。 <b>refresh</b> コマンドは、ファイルの内容に応じて、リストされたデータベースを再ロードする場合と、再ロードしない場合があります。
<b>traceson</b>	サブシステム、サブシステムのグループ、あるいはサブサーバーのトレースを使用可能にします。
<b>tracesoff</b>	サブシステム、サブシステムのグループ、あるいはサブサーバーのトレースを使用不可にします。
<b>lssrc</b>	サブシステム、サブシステムのグループ、あるいはサブサーバーのトレースの状況を取得します。

## フラグ

<b>-b</b>   <b>-c</b> <i>ConfFile</i>	代替構成ファイルを指定します。
<b>-d</b> <i>DebugLevel</i>	デバッグ・オプションを提供します。 <b>-d</b> フラグで、 <b>named8</b> デーモンは、デバッグ情報をデフォルトでファイル <b>/var/tmp/named.run</b> に書き込みます。 <i>DebugLevel</i> 変数は、表示されるメッセージのレベルを指定します。1 から 11 までのレベルが有効であり、レベル 11 は情報量が最も多くなります。
<b>-p</b> <i>PortNumber</i>	<b>named8</b> デーモンが DOMAIN 要求を listen するインターネット・ソケットを再割り当てします。この変数が指定されていないと、 <b>named8</b> デーモンは <b>/etc/services</b> ファイルに定義されたソケットを listen します。 <b>/etc/services</b> ファイルのエントリは domain で始まります。
<b>-w</b> <i>WorkingDirectory</i>	<b>named8</b> デーモンの作業ディレクトリーを変更します。このオプションは、「ディレクトリー」構成オプションによって指定したり、オーバーライドをすることができます。
<b>-t</b> <i>RootDirectory</i>	<b>chroot</b> コマンドを使って、 <b>named8</b> デーモンの新しいルート・ディレクトリーになるディレクトリーを指定します。
<b>-q</b>	すべてのネーム・サービス照会のログを使用可能にします。
<b>-r</b>	サーバーのローカル・データベース外の照会を再帰し、解決するサーバーの機能を使用不可にします。
<b>-f</b>	バックグラウンド・ジョブになるのではなく、フォアグラウンド内のネーム・サーバー・デーモンを実行することを示します。

## シグナル

次のシグナルは、**kill** コマンドを使用して **named8** デーモン処理に送信されると、指定した働きをします。

<b>SIGHUP</b>	<b>named8</b> デーモンは <b>/etc/named.conf</b> ファイルを再読み取りします。 <b>SIGHUP</b> シグナルは、ファイルの内容に応じて、リストされたデータベースを再ロードする場合と再ロードしない場合があります。
<b>SIGILL</b>	統計データを <b>named.stats</b> にダンプします。統計データはファイルに付加されます。

<b>SIGINT</b>	<b>named8</b> デーモンは、現行データベースをファイル <code>/var/tmp/named_dump.db</code> にダンプします。  ダンプ・ファイルでは、ラベル <b>name error</b> の付いた名前は、負のキャッシュ・エントリーを示します。これは、指定したドメイン名が存在しないことをサーバーが応答する場合に発生します。 <b>data error</b> のラベルが付いた名前も負のキャッシュ・エントリーを示します。これは、(有効な) ドメイン名に対して指定したタイプのレコードが存在しないことを、サーバーが応答する場合に発生します。
<b>SIGUSR1</b>	<b>named8</b> デーモンはデバッグの実行を開始します。それ以降の <b>SIGUSR1</b> シグナルはそれぞれデバッグ・レベルを増加させます。デバッグ情報は <code>/var/tmp/named.run</code> ファイルに書き込まれます。
<b>SIGUSR2</b>	<b>named8</b> デーモンは、デバッグの実行を終了します。

## 例

1. **named8** デーモンを通常どおり開始するには、以下のように入力します。

```
startsrc -s named
```

このコマンドは、デーモンを始動します。このコマンドは、**rc.tcpip** ファイル内、またはコマンド行上で使用できます。 **-s** フラグは、後に続くサブシステムを始動することを指定します。 **named8** デーモンのプロセス ID は、始動時に `/etc/named.pid` ファイルに格納されます。

2. **named8** デーモンを通常通り停止するには、以下のように入力します。

```
stopsrc -s named
```

このコマンドはデーモンを停止します。 **-s** フラグは、後に続くサブシステムを停止することを指定します。

3. **named8** デーモンから簡略状況を求めるには、以下のように入力します。

```
lssrc -s named
```

このコマンドにより、デーモン名、デーモンのプロセス ID、デーモンの状態 (アクティブか、非アクティブか) が戻されます。

4. **named8** デーモンのデバッグを可能にするには、以下のように入力します。

```
traceson -s named
```

または

```
kill -30 `cat /etc/named.pid`
```

**named8** デーモンは、上記のいずれかのコマンドに対応してデバッグを実行します。その後の各コマンドはデバッグ・レベルを増加します。デバッグ情報は `/var/tmp/named.run` ファイルに書き込まれます。

5. **named8** デーモンのデバッグをオフにするには、以下のように入力します。

```
tracesoff
```

または

```
kill -31 `cat /etc/named.pid`
```

上記のいずれかのコマンドによって、すべてのデバッグが即座に終了します。

6. **startsrc** コマンドを使用して、最高のデバッグ・レベルで **named8** デーモンを始動するには、以下のように入力します。

```
startsrc -s named -a -d11
```

このコマンドは、デバッグ・メッセージを `/var/tmp/named.run` ファイルに書き込みます。

## ファイル

<code>/usr/sbin/named8</code>	<b>named8</b> デーモンが入っています。
<code>/usr/sbin/named8-xfer</code>	スレーブ・ネーム・サーバーのインバウンド・ゾーン転送の機能を提供します。
<code>/etc/named.conf</code>	いくつかの基本的な動作、ロギング・オプション、およびローカル・データベースの位置などといった、 <b>named8</b> デーモンの構成を指定します。
<code>/etc/resolv.conf</code>	ドメイン・ネーム・サービスの使用方法を指定します。
<code>/etc/rc.tcpip</code>	システムを再始動するたびにデーモンを初期化します。
<code>/etc/named.pid</code>	プロセス ID を保管します。
<code>/etc/services</code>	ソケット・サービスの割り当てを定義します。
<code>/usr/samples/tcpip/named.conf</code>	サンプルの <b>named.conf</b> ファイルがその使い方の説明と一緒に入っています。
<code>/usr/samples/tcpip/named.data</code>	サンプルの DOMAIN データ・ファイルが、その使用上の指示と一緒に入っています。
<code>/usr/samples/tcpip/hosts.awk</code>	<b>/etc/hosts</b> ファイルを <b>/etc/named.data</b> ファイルに変換するためのサンプル <b>awk</b> スクリプトが入っています。このファイルにはその使い方の説明も入っています。
<code>/usr/samples/tcpip/addrns.awk</code>	<b>/etc/hosts</b> ファイルを <b>/etc/named.rev</b> ファイルに変換するためのサンプル <b>awk</b> スクリプトが入っています。このファイルにはその使い方の説明も入っています。

## 関連情報

**nslookup** コマンド、**traceroute** コマンド、**kill** コマンド、**chroot** コマンド。

**named.conf** ファイル・フォーマット、**DOMAIN Cache** ファイル・フォーマット、**DOMAIN Data** ファイル・フォーマット、**DOMAIN Reverse Data** ファイル・フォーマット、**DOMAIN Local Data** ファイル・フォーマット、**resolv.conf** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ の TCP/IP のネーム・レゾリューションのセクション、および TCP/IP デーモンのセクション。

ネットワークおよびコミュニケーションのマネージ のネーム・サーバー・レゾリューションのセクション、およびドメイン・ネーム・レゾリューションの計画のセクション。

---

## named9 デーモン

### 目的

インターネット・ドメイン・ネーム・サーバー。

### 構文

```
named9 [ -c config-file ] [ -d debug-level ] [ -f ] [ -g ] [ -n #cpus ] [ -p port ] [ -s ] [ -v ] [ -x cache-file ]
```

## 説明

**named9** はドメイン・ネーム・システム (DNS) サーバーであり、ISC の BIND 9 で配布されます。DNS の詳細については、RFC 1033、1034、および 1035 を参照してください。引数なしで呼び出された場合、**named** はデフォルト構成ファイル **/etc/named.conf** を読んで、初期データを読み取り、照会を **listen** します。

**dhcpremove8** および **dhcpaction8** スクリプトを **nsupdate** と一緒に使用して、**named9** に対して動的な更新を行うことができます。これを行うには、以下のステップを実行します。

- **nsupdate** を **nsupdate9** に再リンクする。

```
ln -fs /usr/sbin/nsupdate9 /usr/sbin/nsupdate
```

- **dhcpaction8** および **dhcpremove8** スクリプトに対して、以下の変更を行う。

以下の行を：

```
/usr/sbin/nsupdate8 > /dev/null 2>&1
```

次のようにする。

```
/usr/sbin/nsupdate > /dev/null 2>&1
```

## フラグ

<b>-c</b> <i>config-file</i>	デフォルトの <b>/etc/named.conf</b> ではなく、 <i>config-file</i> を構成ファイルとして使用します。構成ファイル内で使用可能なディレクトリー・オプションにより、サーバーが作業ディレクトリーを移動した後も、構成ファイルを再ロードして作業が継続できるようにするため、 <i>config-file</i> には絶対パス名を指定してください。
<b>-d</b> <i>debug-level</i>	デーモンのデバッグ・レベルを <i>debug-level</i> にセットします。 <b>named</b> からのデバッグ・トレースは、デバッグ・レベルが増すほど、冗長になります。
<b>-f</b>	サーバーをフォアグラウンドで実行します (つまり、デーモン化しません)。
<b>-g</b>	サーバーをフォアグラウンドで実行し、すべてのロギングを強制的に <b>stderr</b> に行ないます。
<b>-n</b> <i>#cpus</i>	複数の CPU を活用するため、 <i>#cpus</i> ワーカー・スレッドを作成します。指定しないと、 <b>named</b> は存在する CPU の数を判定し、CPU ごとに 1 つのスレッドを作成します。CPU の数が判定できない場合は、1 つのワーカー・スレッドを作成します。
<b>-p</b> <i>port</i>	ポート <i>port</i> 上の照会を <b>listen</b> します。指定しないと、デフォルトはポート 53 です。
<b>-s</b>	終了時に <b>stdout</b> にメモリー使用統計を書き込みます。 <b>注:</b> このオプションは主に BIND 9 開発者用であり、将来のリリースでは除去または変更される可能性があります。
<b>-v</b>	バージョン番号と出口をレポートします。
<b>-x</b> <i>cache-file</i>	データを <i>cache-file</i> からデフォルト表示のキャッシュにロードします。

**重要:** このオプションは使用しないでください。これは BIND 9 開発者専用です。

## シグナル

通常の操作では、**nameserver** を制御するにはシグナルを使用するのではなく、**rndc** を使用してください。

<b>SIGHUP</b>	サーバーを強制的に再ロードします。
<b>SIGINT、SIGTERM</b>	サーバーをシャットダウンします。



上記以外のシグナルをサーバーに送信した結果は、不明です。

## 構成

**named9** 構成ファイルの完全な説明は、「BIND 9 Administrator Reference Manual」にあります。

## ファイル

`/usr/sbin/named9`  
`/etc/named.conf`  
`/etc/named.pid`

**named9** デーモンが入っています。  
デフォルトの構成ファイル。  
デフォルトの `process-id` ファイル。

## 関連情報

**named8** デーモン。

**named.conf** ファイル・フォーマット、

RFC 1033、RFC 1034、RFC 1035、`rndc(8)`、および `lwresd(8)`。

「BIND 9 Administrator Reference Manual」。

---

## namerslv コマンド

### 目的

システム構成データベースにある、ローカル resolver ルーチンに関するドメイン・ネーム・サーバーのエントリーを直接操作します。

### 構文

#### ネーム・サーバー・エントリーを追加する

```
namerslv -a { -i IPAddress | -D DomainName | -S SearchList }
```

#### ネーム・サーバー・エントリーを削除する

```
namerslv -d { -i IPAddress | -n | -l }
```

#### すべてのネーム・サーバー・エントリーを削除する

```
namerslv -X [ -l ]
```

#### ネーム・サーバー・エントリーを変更する

```
namerslv -c DomainName
```

#### ネーム・サーバー・エントリーを表示する

```
namerslv -s [ -l | -n | -l ] [ -Z ]
```

#### 構成データベース・ファイルを作成する

```
namerslv -b [ -i IPAddress [ -D DomainName ] [ -S SearchList ] ]
```

## 構成データベース・ファイルの名前を変更する

**namerslv -E** *FileName*

## 構成データベース・ファイルを移動してネーム・サーバーの使用を防止する

**namerslv -e**

## 構成データベース・ファイルへファイルをインポートする

**namerslv -B** *FileName*

## 検索リスト・エントリーを変更する

**namerslv -C** *Search List*

## 説明

**namerslv** 低水準コマンドは、システム構成データベースにある、ローカル resolver ルーチンに関するドメイン・ネーム・サーバーのエントリーを追加または削除します。 デフォルトでは、システム構成データベースは **/etc/resolv.conf** ファイルに入っています。

ネーム・サーバーを使うには、以下の 2 つの方法があります。

- システム構成データベースとして使うファイル名を指定します。
- IP プロトコル・アドレスおよび、オプションとしてドメイン名を指定します。

**namerslv** コマンドは、システム構成データベース内の 1 つまたはすべてのドメイン・ネーム・サーバーのエントリーを表示できます。また **namerslv** コマンドは、ネーム・サーバーの使用を中止するために **/etc/resolv.conf** ファイルの名前を変更できます。

ドメイン・ネーム・サーバーのエントリーには以下の 3 つのタイプがあります。

- ローカル・インターネット・ドメインの名前を識別するドメインのエントリー。
- ローカル・ドメイン用のドメイン・ネーム・サーバーの IP アドレスを識別するネーム・サーバーのエントリー。アドレスはドット 10 進数形式でなければなりません。
- ホスト名を解決する際に検索対象となるすべてのドメインをリストする検索リストのエントリー。このリストは、スペースで区切られています。

システム構成データベース内には、1 つのドメインのエントリーおよび最高 3 つまでのネーム・サーバーのエントリーを入れることができます。**/usr/include/resolv.h** ファイル内の **MAXNS** グローバル変数は、ネーム・サーバーの最大数を定義します。1 つの検索エントリーだけを入れることができます。

このコマンドは、Web-based System Manager ネットワーク・アプリケーション (**wsm network** 高速パス) を使用して実行できます。また、System Management Interface Tool (SMIT) **smit namerslv** 高速パスを使用しても実行できます。

## フラグ

- a** システム構成データベースにエントリーを追加します。 **-a** フラグは、**-i** または **-D** フラグと一緒に使用しなければなりません。
- B** *FileName* *FileName* 変数で指定されたファイルから **/etc/resolv.conf** ファイルを復元します。

- b** `/etc/resolv.conf.sv` ファイルを使用してシステム構成データベースを作成します。  
`/etc/resolv.conf.sv` ファイルがない場合は、エラーが戻されます。  
注: `/etc/resolv.conf.sv` ファイルは、出荷されるシステムに付属していません。 ユーザーは、**-b** フラグを使用する前にこのファイルを作成しなければなりません。
- C** `/etc/resolv.conf` ファイルで検索リストを変更します。
- c** *DomainName* システム構成データベース内でドメイン名を変更します。
- D** コマンドがドメイン名のエントリーを処理することを示します。
- d** システム構成データベース内でエントリーを削除します。このフラグは、**-i** *IPAddress* フラグまたは **-n** フラグとともに使用しなければなりません。 **-i** フラグはネーム・サーバーのエントリーを削除します。 **-n** フラグはドメイン名のエントリーを削除します。
- E** *FileName* ネーム・サーバーの使用を中止できるように、システム構成データベース・ファイルの名前を変更します。 `/etc/resolv.conf` ファイルを *FileName* 変数で指定されたファイルに移動します。
- e** `/etc/resolv.conf` ファイルを `/etc/resolv.conf.sv` ファイルに移動して、ネーム・サーバーを使用しないようにします。
- I** (i の大文字) は、**-s** フラグまたは **-X** フラグがネーム・サーバーのエントリーをすべて表示することを指定します。
- i** *IPAddress* コマンドがネーム・サーバーのエントリーを処理することを示します。 IP アドレスの指定にはドット 10 進数フォーマットを使用します。
- l** (L の小文字) は、操作が検索リストに関するものであることを指定します。このフラグと一緒に **-d** フラグと **-s** フラグを使用します。
- n** 操作がドメイン名に関するものであると指定します。このフラグと一緒に **-d** フラグと **-s** フラグを使用します。
- S** *SearchList* システム構成データベース内で検索リストを変更します。
- s** 構成システム・データベース内のすべてのドメイン名のエントリーとネーム・サーバーのエントリーを示します。 **-i** フラグを使用すると、`namerslv` コマンドはネーム・サーバーのエントリーをすべて示します。 **-n** フラグを使用すると、`namerslv` コマンドはデータベース内で検出されたドメイン名のエントリーを示します。
- X** データベース内のすべてのエントリーを削除します。すべてのネーム・サーバーのエントリーを削除するには、このフラグと一緒に **-I** フラグを使用します。
- Z** コロン形式による照会出力を生成します。 `namerslv` コマンドが SMIT 使用可能インターフェースから呼び出された場合は、このフラグを使用します。

## 例

1. ドメイン名 `abc.aus.century.com` を持つドメインのエントリーを追加するには、次のように入力します。  

```
namerslv -a -D abc.aus.century.com
```
2. `abc.aus.century.com` ドメインのエントリーをドメイン名 `xyz.aus.century.com` に変更するには、以下のように入力します。  

```
namerslv xyz.aus.century.com
```
3. IP アドレスが `192.9.201.1` のネーム・サーバーのエントリーを追加するには、次のように入力します。  

```
namerslv -a -i 192.9.201.1
```
4. ローカル resolver ルーチンが使用するドメイン・ネーム・サーバー情報に関して、すべてのシステム構成データベースのエントリーを表示するには、次のように入力します。  

```
namerslv -s
```

以下のフォーマットで出力が表示されます。

```
domain xyz.aus.century.com
name server 192.9.201.1
```

5. **/etc/resolv.conf** ファイルを名前変更してネーム・サーバーの使用を停止し、新しいファイル名 **/etc/resolv.back** を指定するには、次のように入力します。

```
namerslv -E /etc/resolv.back
```

## ファイル

<b>/usr/sbin/namerslv</b>	<b>namerslv</b> コマンドが入っています。
<b>/etc/resolv.conf</b>	デフォルトのシステム構成データベースが入っています。
<b>/etc/resolv.conf.sv</b>	古いシステム構成データベースが入っています。

## 関連情報

**chnamsv** コマンド、**lsnamsv** コマンド、**mknamsv** コマンド、**nslookup** コマンド、**rmnamsv** コマンド、**traceroute** コマンド。

ネットワークおよびコミュニケーションのマネージ の命名のセクション、および TCP/IP デーモンのセクション。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

---

## ncheck コマンド

### 目的

i ノード番号からパス名を生成します。

### 構文

```
ncheck [ [ [ -a ] [ -i InNumber ... ] ] ] [ -s ] [ FileSystem ]
```

### 説明

**ncheck** コマンドは、ファイルシステムのファイルの i ノード番号とパス名を表示します。このコマンドは、パス名に疑問符 (??) を表示して、検出されなかったコンポーネントを示します。名前の始めが ... (省略記号) で表示されるパス名は、ループまたは 11 以上のエントリーからなるパス名を示します。**ncheck** コマンドは単純なハッシュ・アルゴリズムを使用して、表示するパス名を作成し直します。このため、**ncheck** コマンドを実行できるファイルシステムはディレクトリーのエントリーが 50,000 未満のファイルシステムに制限されます。

### フラグ

<b>-a</b>	.(ドット) および ..(ドット・ドット) のファイル名をリストします。
<b>-i InNumber</b>	<i>InNumber</i> パラメーターで指定されたファイルのみをリストします。
<b>-s</b>	スペシャル・ファイルとセット・ユーザー ID モードを持つファイルのみをリストします。

## 例

1. デフォルト・ファイルシステム内の各ファイルの、*i* ノード番号とパス名をリストするには、次のように入力します。

```
ncheck
```

2. 指定されたファイルシステム内のすべてのファイルをリストするには、次のように入力します。

```
ncheck -a /
```

ここでは、各ディレクトリー内の `.` (ドット) と `..` (ドット・ドット) エントリーを含む、`/` (ルート) ファイルシステム内の各ファイルの *i* ノード番号とパス名がリストされます。

3. *i* ノード番号がわかっているファイル名をリストするには、以下のように入力します。

```
ncheck -i 690 357 280 /tmp
```

ここでは、*i* ノード番号が 690、357、または 280 になっている、`/tmp` ファイルシステム内のすべてのファイルの *i* ノード番号とパス名がリストされます。ファイルに複数のリンクがあると、そのすべてのパス名がリストされます。

4. スペシャル・ファイルとセット・ユーザー ID ファイルをリストするには、次のように入力します。

```
ncheck -s /
```

これは、`/` (ルート) ファイルシステム内でスペシャル・ファイル (装置ファイルとも呼ばれる) か、もしくはセット・ユーザー ID モードが使用可能になっているすべてのファイルの *i* ノードおよびパス名をリストします。

## 関連情報

**fsck** コマンド、**sort** コマンド。

オペレーティング・システムおよびデバイスのマネージのファイルシステムのセクション。

---

## nddctl コマンド

### 目的

ネットワーク・デバイス・ドライバー (NDD) に対してコマンドを出します。

### 構文

```
nddctl { -r } Device
```

### 説明

**nddctl** コマンドにより、ユーザーは実行時に NDD デバイスを制御することができます (つまり、ネットワーク接続の中断を伴う、デバイス・ドライバーの再構成を行う必要がありません)。

## フラグ

- r** NDD デバイスに対し、実行時にリンク属性 (スピードおよび二重) を再ネゴシエーションするよう強制します。
- 注:** リンクの再ネゴシエーションを強制すると、デバイスのリセットを必要とします。これにより、デバイスが自身を再初期化する間の数秒間にわたり、ネットワーク接続が失われることがあります。

## パラメーター

*Device* 指定されたコマンドを実行する対象となる NDD デバイスを指定します。

## 終了状況

- 0** コマンドは正常に実行されました。
- >0** エラーが発生しました。

## 例

1. デバイス `ent0` に対し、実行時にリンク属性を再ネゴシエーションするよう強制するには、次のように入力します。

```
nddctl -r ent0
```

## 位置

`/usr/sbin`

---

## ndp コマンド

### 目的

IPv6 の隣接ディスカバリーを表示して制御します。

### 構文

```
ndp [ -n ] hostname
```

```
ndp [ -n ] -a
```

```
ndp [ -d ] hostname | IpAddress
```

```
ndp [ -i interface_index ] -s hostname addr [ temp ]
```

### 説明

**ndp** プログラムは、IPv6 NDP (隣接ディスカバリー・プロトコル) によって使用される IPv6-to-Ethernet、IPv6-to-TokenRing、または IPv6-to-InfiniBand アドレス変換テーブルを表示し、修正します。

フラグを指定しなければ、プログラムは、*hostname* の現在の **ndp** エントリーを表示します。ホストは、IPv6 テキスト表記を使用して、名前や番号で指定されることがあります。

## フラグ

- a 現在の **ndp** エントリーをすべて表示します。
- d スーパーユーザーが **-d** フラグを使用して *hostname* と呼ばれるホストのエントリーを削除できるようにします。
- i *interface\_index* **-s** フラグ (ローカル・リンク・インターフェースで有用) と共に **ndp** エントリーを追加する場合に使用するインターフェースの索引を指定します。
- n ネットワーク・アドレスを数字で表示します (通常、**ndp** は、アドレスを記号で表示しようとします)。
- s *hostname addr* ハードウェア・アドレス *addr* で *hostname* の **ndp** エントリーを作成します。ハードウェア・アドレスはコロンで区切られた 6 つの 16 進バイトで指定されます。コマンド内に **temp** が指定されていない限り、このエントリーは永続となります。

## 例

- a フラグの出力例を示します。

```
# ndp -a
e-crankv6 (:::903:9182) at link#2 0:20:af:db:b8:cf
e-crankv6-11 (fe80:0:100::20:afdb:b8cf) at link#2 0:20:af:db:b8:cf
e-crankv6-11 (fe80::2:c903:1:1e85) at link#5 SQP:0xe SLID0x49 DQP:0x48 DLID:0xf 0:48:fe80::2:c903:1:1e85 [InfiniBand]
# ndp -d e-crankv6-11
e-crankv6-11 (fe80:0:100::20:afdb:b8cf) deleted
# ndp -d fe80::2:c903:1:1e85
```

## 関連情報

**ifconfig** コマンド、**ndpd-host** コマンド、**ndpd-router** コマンド、**autoconf6** コマンド。

---

## ndpd-host デーモン

### 目的

ホストに対する NDP デーモンです。

### 構文

```
ndpd-host [ -d] [ -v] [ -t]
```

### 説明

**ndpd-host** コマンドは、ルーター・ディスカバリー、接頭部ディスカバリー、パラメーター・ディスカバリー、およびリダイレクトなどの非カーネル・アクティビティーに関して隣接ディスカバリー・プロトコル (NDP) を管理します。**ndpd-host** コマンドは、デフォルト・ルーター、デフォルト・インターフェース、およびデフォルト・アドレスなどを含むデフォルト経路を処理します。

### インターフェース

**ndpd-host** コマンドによって、IEEE および CTI Point-to-Point インターフェースが認識されます。

**ndpd-host** コマンドによって、既知のインターフェースのすべてのパケットがリンク・ローカル・アドレスと交換されます。インターフェースの状況のどのような変更も検出されます。インターフェースがダウンしたり、あるいはそのリンク・ローカル・アドレスを失うと、NDP 処理はそのインターフェース上で停止します。インターフェースが立ち上がると、NDP 処理が開始します。

IEEE インターフェースは、**autoconf6** コマンドを使用して構成されます。PPP インターフェースは、**pppd** デーモンを使用して構成されます。トークン・ネゴシエーションはリンク・ローカル・アドレスを定義します。CTI が構成されたトンネルを介して Router Advertisement (ルーター通知) を送信するためには、ローカルおよびリモートのリンク・ローカル・アドレスを持っていないければなりません。

注：up のすべての Point-to-Point インターフェースの場合、**ndpd-host** は、ローカル・アドレスの lo0 を介してローカル経路指定を設定します。

## フラグ

<b>-d</b>	デバッグを可能にします (例外条件およびダンプ)。
<b>-v</b>	関心のあるイベントすべてのログ記録を取ります (デーモン情報およびコンソール)。
<b>-t</b>	それぞれのログにタイム・スタンプを追加します。

## シグナル

<b>SIGUSR1</b>	詳細をオンにします。
<b>SIGUSR2</b>	詳細をオフにします。
<b>SIGINT</b>	ndpd-host の現在の状態を syslog や stdout にダンプします。
<b>SIGTERM</b>	ndpd-host をクリーンアップして終了します。

## 関連情報

**ifconfig** コマンド、**route** コマンド、**autoconf6** コマンド、および **ndpd-router** コマンド。

---

## ndpd-router デーモン

### 目的

ルーター用の NDP および RIPng デーモンです。

### 構文

```
ndpd-router [ -r ] [ -p ] [ -M ] [ -O ] [ -s ] [ -q ] [ -g ] [ -n ] [ -R ] [ -S ] [ -d ] [ -t ] [ -v ] [ -H ] [ -m ]  
[ -u port ] [ -D max[min[life]] ] [ -P [invlife]/[deplife] ] [ -T [reachtim]/[retrans]/[hlim] ]
```

### 説明

**ndpd-router** デーモンは、非カーネル・アクティビティに関して隣接ディスカバリー・プロトコル (NDP) を管理します。これはルーター返信請求 (Router Solicitation) を受信し、ルーター通知 (Router Advertisement) を送信します。また、RIPng プロトコルを使用してルーティング情報を交換することもできます。

**/etc/gateway6** ファイルには、**ndpd-router** のオプションが用意されています。このファイルは、プログラムの実行中に修正できます。これらの変更は、メッセージの出力または受信前、あるいは、HUP シグナルの受信時に検査されます。このファイルには、ディレクティブが 1 行に 1 つずつの割合で (コメントとして # が付けられて) 収められています。このファイルでは、すべての IPv6 アドレスと接頭部を数字フォーマットにする必要があります。シンボル名は使用できません。ゲートウェイ・ディレクティブを除き、各行はキーワードで始まり、*key=argument* の形式を持つオプションからなります。



## インターフェース

**ndpd-router** デーモンによって、IEEE および CTI Point-to-Point インターフェースが認識されます。

**ndpd-router** デーモンによって、既知のインターフェースのすべてのパケットがリンク・ローカル・アドレスと交換されます。インターフェースの状況のどのような変更も検出されます。インターフェースがダウンしたり、あるいはそのリンク・ローカル・アドレスを失うと、NDP と RIPng の処理は、そのインターフェース上で停止します。インターフェースが立ち上がると、NDP と RIPng の処理は開始されます。

ルーター通知または RIPng パケット (またはその両方) を送信するには、ローカルおよび リモートの両方のリンク・ローカル・アドレスを構成する必要があります。

## フラグ

<b>-H</b>	モバイル IPv6 ホーム・エージェントとして機能するために必要な NDP フィーチャーを、システムが処理できるようにします。
<b>-m</b>	モバイル IPv6 モバイル・ノードの移動検出をシステムが補助できるようにします。
<b>-D</b> <i>max</i> [ <i>min</i> / <i>life</i> ]	非送信請求ルーター通知を <i>min</i> 秒から <i>max</i> 秒の間隔で送信します。デフォルトの <i>max</i> 値は 600 秒で、有効範囲は 4 から 1800 秒です。デフォルトの <i>min</i> は、 <i>max</i> / 3 で、有効範囲は 1 から $0.75 * max$ です。ルーターのライフタイムは、 <i>life</i> で設定されます。このデフォルト値は $10 * max$ です。有効範囲は 0 から 65535 秒です。
<b>-T</b> [ <i>reachtim</i> ] / [ <i>retrans</i> ] / [ <i>hlim</i> ]	<i>reachim</i> がゼロでなければ、BaseReachableTime フィールドを <i>reachim</i> 秒に設定します。 <i>retrans</i> がゼロでなければ、RetransTime フィールドを <i>retrans</i> 秒に設定します。 <i>hlim</i> がゼロでなければ、ルーター通知のホップ制限フィールドを <i>hlim</i> に設定します。
<b>-M</b>	通知の <b>M</b> フラグ (状態構成) を設定します。
<b>-O</b>	通知の <b>O</b> フラグ (その他の状態情報) を設定します。
<b>-p</b>	接頭部 (インターフェース構成から入手) は提供しません。
<b>-P</b> [ <i>invlife</i> ]/[ <i>deplife</i> ]	告知済み接頭部に関して、無効な存続値と使用すべきではない存続値を (秒数で) 設定します。デフォルト値は 0xffffffff (無限) です。
<b>-r</b>	ルーター通知でデフォルト・ルーターを提供しません。
<b>-s</b>	RIPng プロトコルを使用可能にします (デフォルトの設定は、RIPng の使用不可です)。
<b>-q</b>	RIPng プロトコルを使用可能にします。ただし、RIPng パケットは送信しません。
<b>-g</b>	RIPng でデフォルト経路指定をブロードキャストします。
<b>-n</b>	RIPng で受信した経路指定はインストールしません。
<b>-u</b> <i>port</i>	UDP ポートの <i>port</i> を RIPng に使用します。デフォルトは 521 です。
<b>-R</b>	split horizon without corrupting reverse を RIPng に使用します。
<b>-S</b>	どのような split horizon (スプリット・ホライズン) も RIPng に使用しません。
<b>-d</b>	デバッグを可能にします (例外条件およびダンプ)。
<b>-v</b>	関心のあるイベントすべてのログ記録を取ります (デーモン情報およびコンソール)。
<b>-t</b>	ログに記録されたメッセージにタイム・スタンプを追加します。

## 使用可能なディレクティブ

**/etc/gateway6** ファイルの主なディレクティブは以下のものです。

**option** [*option-directive* ...]

per-interface/default オプションを設定します。

**prefix** [*prefix-directive* ...]

per-interface/default プレフィックス処理オプションを設定します。

**filter** [*filter-directive* ...]

per-interface/default フィルターを設定します。

## ゲートウェイ・ディレクティブ

RIPng パケットまたはカーネルに経路を設定します。

これらのディレクティブについては、以下で詳細に説明します。

## オプションのディレクティブ

別の per-interface オプションを設定します。

**if** オプションの後に続けて **option** ディレクティブへの値を設定する場合は、リスト内にコンマで区切って指定する必要があります。

**注:** 以下の **option** ディレクティブには、少なくとも 1 つのオプション (**if** オプション以外) を指定する必要があります。 **if** オプションを指定する場合は、 **option** ディレクティブの後に最初のオプションとして指定する必要があります。 **if** オプションと、これに続くオプションのリスト (コンマで区切られたリスト) の間には 1 つのスペースが必要です。

### 構文:

```
option [ if=n1,n2 ] ripin=(y|n),ripout=(y|n|S|R),rtadv=(y|n|min[/max]),flag=[M|O],life=Seconds,reach=Seconds,retrans=Seconds
```

<b>if</b> = <i>list</i> <b>interface</b> = <i>list</i>	キーワードがない場合、オプション・ディレクティブがデフォルト・オプションになります。インターフェース・フィールドがある場合、オプション・パラメーターは、リストされているインターフェースにのみ適用されます。 <i>list</i> は、コンマで区切られます。 <i>le*</i> を使用してすべての <i>leX</i> インターフェースを指定できます。デフォルト・オプションは、 <i>/etc/gateway6</i> ファイルの 1 行目に指定する必要があります。
<b>mtu</b> [= <i>mtuval</i> ]	ルーター通知で <i>mtuval</i> の MTU 値を通知します。 <i>mtuval</i> 引数がない場合、通知される MTU は、インターフェースの MTU になります。 <i>mtuval</i> が 0 の場合、MTU の通知は抑止されます。
<b>ripin</b> =( <i>n</i>   <i>y</i> )	着信の RIPng パケットの listen は行いません。 RIPng パケットの送信 (send) は行いません。 split horizon (スプリット・ホライズン) を <b>-S</b> フラグでは使用しません。 <b>-R</b> フラグの場合は、split horizon without poisoning reverse を使用します。
<b>rtadv</b> =( <i>n</i>   <i>y</i>   <i>min</i> [ <i>max</i> ])	ルーター通知の送信 (send) は行いません。 <i>min</i> [/ <i>max</i> ] オプションでは、ルーター通知の間隔 (秒) を指定します。
<b>flag</b> ={ <i>M</i>   <i>O</i> }	ルーター通知で状態モード・フラグを設定します。 <b>M</b> 状態構成を使用します。 <b>O</b> 状態構成を使用します。ただし、アドレスには使用しません。
<b>life</b> = <i>Seconds</i>	ルーター通知のルーター存続フィールドの値を (秒単位で) 設定します。
<b>reach</b> = <i>Seconds</i>	ルーター通知の到達可能フィールドの値を (秒単位で) 設定します。
<b>retrans</b> = <i>Seconds</i>	ルーター通知の再伝送間隔フィールドの値を (秒単位で) 設定します。

## 接頭部ディレクティブ

ルーター通知ディレクティブで告知される接頭部を定義します。インターフェースの `prefix-directive` がない場合は、ルーター通知に、インターフェースのアドレス・リストから導出された接頭部のリストが含まれます。`prefix-directive` がある場合は、ルーター通知に、さまざまな接頭部指示によって (順番に) 定義された接頭部のリストが含まれます。カーネルには接頭部はインストールされません。`prefix=none` という形式の接頭部を持つディレクティブが 1 つあれば、接頭部リストは通知されません。

構文:

```
prefix if=n prefix=(none|xxx::/PrefixLength) flag=[L][A] valid=Seconds deprec=Seconds
```

<code>if=Interface</code> または <code>interface=Interface</code>	ディレクティブを適用するインターフェースを指定します。 <code>prefix</code> ディレクティブには <code>if</code> キーワードが必須です。これはオプションではありません。
<code>prefix=xxx::/PrefixLength</code>	通知される接頭部。
<code>flag=[L][A]</code>	接頭部に対して <code>L</code> または <code>A</code> (あるいはその両方の) フラグを設定します (デフォルトは <code>LA</code> です)。
<code>deprec=Seconds</code>	接頭部を使用すべきでない時間を (秒単位で) 指定します。
<code>valid=Seconds</code>	接頭部の妥当な時間を (秒単位で) 指定します。

## フィルター・ディレクティブ

着信 (`filter=in`) または発信 (`filter=out`) RIPng パケットのフィルター・パターンを定義します。1 つのインターフェースにつき着信フィルターと発信フィルターが 1 つずつあり、明示フィルターを使用しないインターフェースに関して、デフォルトの着信フィルターと発信フィルターが 1 つずつあります。

受信した RIPng 情報はすべて、インターフェースの入力フィルターに照らしてテストされ、入力フィルターがない場合は、デフォルトの入力フィルターに照らしてテストされます。静的インターフェース経路指定は、インターフェース、ならびにそのインターフェースのリンク・ローカル・アドレスが指定されたゲートウェイから発信される入力情報として示されます。`gateway` キーワードを持つゲートウェイ・ディレクティブで設定された経路は、指定されたインターフェースとゲートウェイから来た入力情報と見なされます。`ゲートウェイ`・キーワードなしにゲートウェイ・ディレクティブで設定した経路指定とデフォルト経路指定 (`-g` フラグ) は、インターフェースからではなく、ゲートウェイ :: から発信される入力情報として示されます (デフォルトの入力フィルターが適用されます)。

送信される RIPng 情報はすべて、インターフェースの出力フィルターに照らしてテストされ、出力フィルターがない場合は、デフォルトの入力フィルターに照らしてテストされます。

それぞれのフィルターは、突き合わせパターンのシーケンスです。これらのパターンは、順番にテストされます。それぞれのパターンごとに、接頭部の長さ、ソース・ゲートウェイ (入力フィルターの場合)、ならびに、その (ゼロで埋め込まれた) 接頭部が固定接頭部と一致しているかどうかをテストすることができます。パターンに複数のテスト記述が入っている場合、突き合わせは、それらすべてのテストの結合になります。最初の一致するパターンでは、実行するアクションを定義します。一致したパターンがない場合は、デフォルトのアクションが受け入れられます。実行される可能性のあるアクションは、`accept`、`reject`、および `truncate/NumberOfBits` です。`truncate/NumberOfBits` アクションは、パターンが一致し、かつ接頭部の長さが `NumberOfBits` 以上である場合に、新しい長さ `NumberOfBits` を持つ接頭部を受け入れることを意味します。受け入れられる接頭部はすぐに受け入れられます。つまり、フィルターに照らして再度検査されることはありません。

例えば、次のディレクティブは、明示的な発信フィルターを持たないインターフェースにホスト経路を送信することを禁止します。

```
filter=out length==128 action=reject
```

構文:

**filter**=(in|out) [if=*n1,n2*] **prefix**=*xxx::/NumberOfBits* **gateway**=*xxx* **length**=(*=|>|<|<|>*)*NumberOfBits*  
**action**=(**accept|reject|truncate**/*xx*)

<b>if</b> = <i>list</i> または <b>interface</b> = <i>list</i>	interface キーワードがない場合、フィルター・ディレクティブがデフォルト・オプションになります。interface フィールドがある場合、フィルター・パターンは、指定されたすべてのインターフェースのフィルターの最後に追加されます。list は、コンマで区切られます。例えば、 <b>interface</b> =le* と指定すれば、すべての leX インターフェースを指定できます。
<b>prefix</b> = <i>xxx::/NumberOfBits</i>	パターンは、 <i>xxx::/NumberOfBits</i> が RIPng パケット内の接頭部の接頭部である場合のみ一致します。
<b>gateway</b> = <i>xxx</i>	パターンは、RIPng メッセージがソース・アドレス <i>xxx</i> から来て、着信フィルターの中にある場合のみ一致します。
<b>length</b> =( <i>= &gt; &lt; &lt; &gt;</i> ) <i>NumberOfBits</i>	パターンは、RIPng メッセージ内の接頭部の長さが <i>NumberOfBits</i> に等しい (または、指定された演算子により、より大、より小などの) 場合にのみ一致します。
<b>action</b> =( <b>accept reject truncate</b> / <i>NumberOfBits</i> )	パターンが一致した場合に実行するアクションを指定します。つまり、メッセージを受け入れること、メッセージをリジェクトすること、あるいは、受け入れるが接頭部を <i>NumberOfBits</i> ビットに切り捨てることを指定します。

## ゲートウェイ・ディレクティブ

ゲートウェイ・ディレクティブを使用すると、RIPng パケットまたはカーネル (あるいはその両方) に経路を設定できます。これらのディレクティブは、**/etc/gateway6** ファイルの最後に、その他のディレクティブの後に指定する必要があります。

構文:

```
xxx::/NumberOfBits metric Value
```

```
xxx::/NumberOfBits metric Value gateway IPv6Address ifname
```

2 番目の構文は、カーネルに経路を追加する場合に使用します。

## 例

次に **/etc/gateway6** ファイルの例を示します。

すべてのアドレスが 5f06:2200:c001:0200:xxxx の形式を持つサイトで、次の例は、すべてのサイトを記述する 1 つの経路だけが、すべての構成済みトンネル・インターフェース (CTI) **ctiX** インターフェースにエクスポートされることを意味します。示されているキーワードの省略形は有効です。

```
filt=out if=cti* pref=5f06:2200:c001:0200::/64 len=>=64 act=trunc/64
```

デフォルトの発信経路を設定します。

```
::/0 metric 2 gateway 5f06:2200:c102:0200::1 cti0
```

RIPng を持つアクティブなすべての CTI インターフェースが、デフォルト経路を定義することを宣言します。

```
filter=in if=cti* act=trunc/0
```

次の例は、外部への接続として cti0 を使用するサイトを定義している例です。このとき、cti0 は ctiX を介して接続される他のサイトを集約しており、また、split horizon without corrupting reverse 方式を使用しています。

```
option if=cti* ripout=R
filter=out if=cti0 prefix=5f06:2200::/24 len=>=24 act=trunc/24
filt=out if=cti* pref=5f06:2200:c001:0200::/64 len=>=64 act=trunc/64
filter=in if=cti0 act=trunc/0
filter=in if=cti* prefix=5f06:2200::/24 len=>=24 act=trunc/64
filter=in if=cti* act=reject
```

## 診断

すべてのエラーは、デバッグ・オプションが設定されていない場合、**daemon.err** レベルでログが取られます。これには、**/etc/gateway6** ファイル内のすべての構文エラー、ならびに、異なるルーター相互間の構成の不一致が含まれます。

## シグナル

**ndpd-router** は次のシグナルに応答します。

<b>SIGINT</b>	syslog が定義されていれば、その現在の状態を syslog にダンプします。定義されていなければ、stdout にダンプします。
<b>SIGHUP</b>	ファイル <b>/etc/gateway6</b> が再度読み取られます。
<b>SIGUSR1</b>	詳細度を増大します。
<b>SIGUSR2</b>	詳細度をリセットします。
<b>SIGTERM</b>	適切な状態にリセットし、停止します。
<b>SIGQUIT</b>	適切な状態にリセットし、停止します。

## ファイル

**/etc/gateway6**

## 関連情報

**ifconfig** コマンド、**kmodctrl** コマンド、**mobip6reqd** コマンド、**mobip6ctrl** コマンド、**rc.mobip6** コマンド、**route** コマンド、**autoconf6** コマンド、**ndpd-host** コマンド。

ネットワークおよびコミュニケーションのマネージ のモバイル IPv6 のセクション。

---

## ndx コマンド

### 目的

文書の主題ページの索引を作成します。

## 構文

**ndx** [ *SubjectFile* ] " *FormatterCommandLine* "

## 説明

**ndx** コマンドは、主題リスト (*SubjectFile*) が与えられると、指定された英語文書を検索して、主題ページの索引を標準出力に書き出します。

文書には、**mm**、**mmt**、**nroff**、または **troff** コマンドのフォーマットされたディレクティブが入っていないなければなりません。フォーマッター・コマンド行は **ndx** コマンドに、文書の最終バージョンを作成するのに、**troff** コマンド、**nroff** コマンド、**mm** コマンド、または **mmt** コマンドのいずれを使用するのかを伝えます。これらのコマンドは以下を行います。

**troff** または **mmt**      フォーマッターに **troff** コマンドを指定します。  
**nroff** または **mm**      フォーマッターに **nroff** コマンドを指定します。

## パラメーター

*SubjectFile*

索引に含める主題のリストを指定します。各サブジェクトは新しい行から始め、次のフォーマットでなければなりません。

```
word1[word2...][,wordk...]
```

以下に例を示します。

```
printed circuit boards
arrays
arrays, dynamic storage
Smith, W.P.
printed circuit boards, channel-oriented
                                 multi-layer

Aranoff
University of Illinois
PL/1
```

*FormatterCommandLine*

主題は 1 桁目から始まらなければなりません。最終フォーマットの文書を作成します。このパラメーターの構文は以下のとおりです。

```
Formatter [Flag...] File...
```

```
mm -Tlp File(s)
nroff -mm -Tlp -rW60 File(s)
troff -rB2 -Tibm3816 -r01.5i File(s)
```

フォーマッター・コマンド行については、**mm** コマンド、**mmt** コマンド、**nroff** コマンド、および **troff** コマンドを参照してください。

*Flag* 変数によって指定されたフラグは、最終的なフォーマットの文書を表示する際に **troff**、**nroff**、**mm**、または **mmt** コマンドに指定するフラグと同じです。これらのフラグは文書に配置される主題の正しいページ番号を決定するために必要です。**ndx** コマンドは最終バージョンの文書を表示しません。作成者は文書を別に作成しなければなりません。文書が完了し、さらに変更がなくなった後にだけインデクサーを使用します。



## 例

1. 次のコマンドは file ドキュメントの主題ページの索引を作成し、主題を subfile リストから取り出します。

```
ndx subfile "nroff -mm -rW70 file" > indexfile
```

ページ番号は、次のように作成された文書に対応します。

```
nroff -mm -rW70 file
```

2. 次のコマンドは文書 ch1、ch2、および ch3 の主題ページの索引を作成します。

```
ndx subfile "mm -rW60 -rN2 -r00 ch1 ch2 ch3" > indexfile
```

ページ番号は、次のように作成された文書に対応します。

```
mm -rW60 -rN2 -r00 ch1 ch2 ch3
```

3. 次のコマンドは file ドキュメントの主題ページの索引を作成します。

```
ndx Subjfile "troff -rB2 -rW5i -r01.5i -mm file" > indexfile
```

ページ番号は、次のように入力して作成された文書に対応します。

```
troff -rB2 -rW5i -r01.5i -mm file
```

## 関連情報

mm コマンド、mmt コマンド、nroff コマンド、subj コマンド、troff コマンド。

---

## neqn コマンド

### 目的

nroff コマンドの数式テキストをフォーマットします。

### 構文

```
neqn [ -d Delimiter1Delimiter2 ] [ -f Font ] [ -p Number ] [ -s Size ] [ - ] [ File ... | - ]
```

### 説明

neqn コマンドはタイプライター型端末上で数学的なテキストをフォーマットするための nroff プリプロセッサです。neqn コマンドの出力を次のように nroff コマンドにパイプ接続します。

```
neqn [Flag...] File... | nroff [Flag...] | [Printer]
```

neqn コマンドは 1 つ以上のファイルを読み取ります。File パラメーターでファイルを指定しないか - (ハイフン) フラグを最後のパラメーターとして指定した場合、デフォルトの設定として標準入力を読み取ります。.EQ マクロで始まる行により、数式テキストの開始を表します。また、.EN マクロで始まる行により、数式テキストの終了を表します。中央揃えや番号付けなどの追加のフォーマット機能を提供するためマクロ・パッケージに定義できるように、これらの行は nroff コマンドによって変更されません。

- (二重ハイフン) 区切り文字はフラグの終わりを示します。

出力先のデバイスによっては、nroff コマンドでフォーマットされた neqn コマンドは、正しい出力を生成するために、col コマンドによる後処理が必要になることもあります。

**eqn** コマンドにより、使用する入力フォーマットやキーワードに関する詳しい情報が提供されます。

## フラグ

**-d***Delimiter1**Delimiter2*

**.EQ** マクロと **.EN** マクロで囲んだ入力以外に、**neqn** コマンドが処理するテキストの区切り文字として 2 つの ASCII 文字、*Delimiter1* および *Delimiter2* を設定します。これらの区切り文字間のテキストは **neqn** コマンドの入力として扱われます。

ファイル内で **delim** *Delimiter1**Delimiter2* 要求を使用して、**neqn** テキスト用の区切り文字を設定することもできます。これらの区切り文字は **delim off** 要求によってオフにされます。区切り文字に囲まれていないすべてのテキスト、**.EQ** マクロおよび **.EN** マクロは未処理のまま渡されます。

**-f***Font*

**neqn** コマンドで処理したすべてのテキストのフォントを *Font* 変数で指定した値に変更します。*Font* の値 (フォント名または位置) は、1 文字または 2 文字の ASCII 文字でなければなりません。

**-p***Number*

添え字および肩文字のサイズを指定されたポイント数まで下げます。デフォルトは 3 ポイントです。

**-s***Size*

**neqn** コマンドで処理したすべてのテキストのポイント・サイズを *Size* 変数で指定した値に変更します。

-  
—

標準入力から読み取ります。  
(二重ハイフン) フラグの終わりをマークします。

## ファイル

**/usr/share/lib/pub/eqnchar**

特殊キャラクターの定義が入っています。

## 関連情報

**checkeq** コマンド、**col** コマンド、**eqn** コマンド、**mm** コマンド、**nroff** コマンド、**tbl** コマンド。

**.EN** マクロ、**.EQ** マクロ、**mm** マクロ。

**eqnchar** ファイル・フォーマット。

---

## netpmon コマンド

### 目的

アクティビティをモニターし、ネットワークの入出力とネットワークに関連する CPU の使用量の統計情報を報告します。

### 構文

```
netpmon [ -o File ] [ -d ] [ -T n ] [ -P ] [ -t ] [ -v ] [ -r PURR ] [ -O ReportType ... ] [ -i Trace_File ]  
-n Gennames_File ]
```



## 説明

**netpmon** コマンドは、システム・イベントのトレースをモニターし、モニター時間中にネットワークのアクティビティーおよびパフォーマンスについて報告します。デフォルトでは、**netpmon** コマンドはバックグラウンドで実行し、一方で 1 つまたは複数のアプリケーション・プログラムまたはシステム・コマンドが実行およびモニターされます。**netpmon** コマンドは、リアルタイムで自動的にネットワーク関連のシステム・イベントのトレースを始動し、モニターします。デフォルトでは、トレースは即座に開始されます。オプションでユーザーが **trcon** コマンドを発行してからトレースを開始するようにも設定できます。**trcstop** コマンドによりトレースを停止すると、**netpmon** コマンドは指定されたレポートすべてを生成してから終了します。

**netpmon** コマンドはオフライン・モードでも機能しますので、前に生成されたトレース・ファイルでも使用できます。このモードで使用するためには、**gennames** コマンドによって生成されたファイルも必要です。**gennames** ファイルは、トレースが停止した直後に、同一マシン上に生成されているはずですが、オフライン・モードで実行されている場合、**netpmon** コマンドはソケットによって使用されるプロトコルを認識することができないため、これにより、ソケット報告で使用可能な詳しさのレベルが制限されます。

**netpmon** コマンドは以下のシステム実行状況を報告します。

### CPU の使用状況

**netpmon** コマンドは、すべてのスレッドおよび割り込みハンドラーによる CPU の使用状況をモニターします。また、この使用状況のうちどれくらいがネットワーク関連のアクティビティーによるものかを予測します。

### ネットワークのデバイス・ドライバーの入出力

**netpmon** コマンドは、トークンリングとファイバー分散データ・インターフェース (FDDI) のネットワーク・デバイス・ドライバーによる入出力操作をモニターします。送信入出力の場合に、使用率、キューの長さ、および宛先ホストもモニターします。受信 ID については、このコマンドは demux レイヤーの時間もモニターします。

### インターネット・ソケット・コール

**netpmon** コマンドは、インターネット・ソケット内のすべての **send**、**recv**、**sendto**、**recvfrom**、**read**、および **write** サブルーチンをモニターします。以下のそれぞれのプロトコル・タイプについて、プロセス単位で統計情報を報告します。

- インターネット制御メッセージ・プロトコル (ICMP)
- 伝送制御プロトコル (TCP)
- ユーザー・データグラム・プロトコル (UDP)

### NFS の入出力

**netpmon** コマンドは、クライアント・ネットワーク・システム (NFS) ファイル、クライアントの NFS リモート・プロシージャ・コール (RPC) 要求、および NFS サーバーの読み取りまたは書き込み要求に関する **read** および **write** サブルーチンをモニターします。このコマンドは、プロセスごと、またはオプションのスレッドごと、および、各サーバーのファイルごとにサブルーチン統計情報を報告します。**netpmon** コマンドはさらにサーバーごとにクライアントの RPC 統計情報を報告し、クライアントごとにサーバーの読み取りおよび書き込み統計情報を報告します。

コマンド行フラグを使用して、前述のレポート・タイプを組み合わせて指定できます。デフォルトでは、すべてのレポートが作成されます。

**注:** **netpmon** コマンドで作成されるレポートはかなり長くなることがあります。したがって、レポートを出力ファイルに書き込むときには、**-o** フラグを常に使用する必要があります。**netpmon** コマンドは、システム・トレース機能を使用してパフォーマンス・データを入手します。トレース機能が

サポートするのは、1 つの出力ストリームだけです。したがって、一度に 1 つの **netpmon** プロセスまたは **trace** プロセスしか活動化できません。別の **netpmon** プロセスまたは **trace** プロセスを既に実行中の場合、**netpmon** コマンドは次のメッセージで応答します。

```
/dev/systrace: Device busy
```

かなりのネットワーク集中アプリケーションをモニターしている場合、**netpmon** コマンドはトレース・イベントをそれらがリアルタイムで作成されるのと同じ速さで使い尽くせないことがあります。このような状況が発生すると、次のエラー・メッセージ

```
Trace kernel buffers overflowed, N missed entries
```

が標準エラーで表示され、トレース・バッファがいっぱいになった間に失われたトレース・イベントの数が示されます。**netpmon** コマンドは、ネットワーク・アクティビティのモニターを継続しますが、レポートの正確さはある程度損なわれます。オーバーフローを避ける方法の 1 つは、**-T** フラグを使用して、トレース・バッファ・サイズを大きくして、オーバーフローの前にトレース・イベントの大規模なバーストを用意しておくことです。オーバーフローの問題を完全に回避するための別の方法として、**netpmon** をオフライン・モードで実行することもできます。

メモリー制約環境 (メモリーの需要が供給を超えてしまう) で実行しているときは、**-P** フラグを使用して、リアルタイムの **netpmon** プロセスのテキストおよびデータ・ページをメモリー内に固定できます。こうしておくこと、それらのページがスワップアウトされることはありません。**-P** フラグを使用せず、**netpmon** プロセスをスワップアウト可能にしておくこと、**netpmon** コマンドの進行が遅延して、このコマンドがトレース・イベントを十分なスピードで処理できず、トレース・バッファのオーバーフローを防げません。

**/unix** ファイルと実行中のカーネルが同じでないと、カーネル・アドレスが正しくないため、**netpmon** コマンドは終了します。

## フラグ

- d** **netpmon** コマンドは始動しますが、ユーザーが **trcon** コマンドの実行を完了するまでトレースは遅れます。デフォルトでは、トレースを即座に開始します。
- i Trace\_File** ライブ状態のシステムからではなく、**trace** コマンドで生成されたファイル *Trace\_File* からトレース・レコードを読み取ります。トレース・ファイルは、最初に **trcpt -r** コマンドを使用して未加工フォーマットで書き直さなければなりません。このフラグは **-n** フラグと一緒になければ使用することができません。
- n Gennames\_File** **gennames** コマンドで生成されたファイル *Gennames\_File* から必要なマッピング情報を読み取ります。**-i** フラグを使用するときには、このフラグは必須です。
- o File** 標準出力ではなく、指定した *File* にレポートを書き込みます。

- O ReportType ...** 指定したタイプのレポートを作成します。有効なレポート・タイプは以下のとおりです。
- cpu** CPU の使用状況
  - dd** ネットワークのデバイス・ドライバーの入出力
  - so** インターネット・ソケット・コールの入出力
  - nfs** NFS I/O (すべてのバージョン)
  - nfs2** NFS バージョン 2 I/O
  - nfs3** NFS バージョン 3 I/O
  - nfs4** NFS バージョン 4 I/O
  - all** すべてのレポートが作成されます。これはデフォルト値です。
- P** モニター・プロセスをメモリーに固定します。このフラグを指定すると、モニター期間の間は **netpmon** テキストおよびデータのページがメモリー内に固定されます。このフラグは、メモリー制約環境で実行しているときに、リアルタイムの **netpmon** プロセスがメモリー・スペースを超えて実行しないようにするときに使用できます。
- r PURR** パーセントと CPU 時間の計算では、TimeBase ではなく PURR 時間を使用します。経過時間の計算には影響しません。
- t** CPU レポートをスレッドごとに表示します。
- T n** カーネルのトレース・バッファ・サイズを *n* バイトに設定します。デフォルトのサイズは 64000 バイトです。必要であれば、バッファ・サイズを増やして、より大きなイベントのバーストに対応することもできます。(通常のイベント・レコード・サイズは 30 バイトくらいです。)
- 注:** カーネルのトレース・ドライバーはダブル・バッファリングを使用します。したがって、実際には *n* バイトのサイズが割り当てられた 2 つのバッファが存在します。これらのバッファはメモリー内に固定されるので、ページングに左右されることはありません。
- v** すべての情報をレポートに表示します。最大 20 個のアクティブ・プロセスおよびアクティブ・ファイルだけではなく、すべてのプロセスおよびアクセスされたすべてのリモート・ファイルがレポートに入ります。

## レポート

**netpmon** コマンドで作成するレポートはヘッダーから始まります。ヘッダーでは、日付、マシン ID、およびモニター期間の長さ (秒単位) が識別されます。ヘッダーの後に、指定したすべてのレポート・タイプについて 1 組の要約および詳細なレポートが続きます。

### CPU の使用レポート

**Process CPU Usage Statistics (プロセスの CPU 使用統計情報):** 各行にはプロセスに関連した CPU の使用状況について記述されています。詳細オプションを指定しない限り、最大 20 個までの活動化プロセスしかリストされません。レポートの一番下には、すべてのプロセスの CPU 使用状況が合計され、CPU アイドル時間が報告されます。

#### Process

プロセス名

**PID** プロセス ID 番号

#### CPU Time

このプロセスで使用した CPU 時間の合計

**CPU %** このプロセスに関する CPU の使用状況 (合計時間のパーセント)

**Network CPU %**

このプロセスがネットワーク関連コードを実行するのに費やした時間 (合計時間のパーセント)

**Thread CPU Usage Statistics**

**-t** フラグを使用すると、上記の各プロセス行の直後に、そのプロセスが所有する各スレッドの CPU 状況を記述する行が続きます。これらの行のフィールドは、プロセスの場合と同じですが、name フィールドが異なります。(スレッドには名前が付いていません。)

**第 1 レベル割り込みハンドラー使用統計情報:** 各行には、第 1 レベル割り込みハンドラー (FLIH) に関連した CPU の使用状況について記述されています。レポートの一番下には、すべての FLIH の CPU 使用状況が合計されています。

**FLIH** 第 1 レベル割り込みハンドラーの説明

**CPU Time**

この FLIH で使用した CPU 時間の合計

**CPU %** この割り込みハンドラーに関する CPU の使用状況 (合計時間のパーセント)

**Network CPU %**

この割り込みハンドラーがネットワーク関連イベントを実行するのに費やした時間 (合計時間のパーセント)

**2 次レベル割り込みハンドラー:** 各行には、2 次レベル割り込みハンドラー (SLIH) に関連した CPU の使用状況について記述されています。レポートの一番下には、すべての SLIH の CPU 使用状況が合計されています。

**SLIH** 2 次レベル割り込みハンドラーの説明

**CPU Time**

この SLIH で使用した CPU 時間の合計

**CPU %** この割り込みハンドラーに関する CPU の使用状況 (合計時間のパーセント)

**Network CPU %**

この割り込みハンドラーがネットワーク関連イベントを実行するのに費やした時間 (合計時間のパーセント)

## 要約ネットワーク・デバイス・ドライバ・レポート

**ネットワーク・デバイス・ドライバ統計情報 \_ デバイス別:** 各行には、ネットワーク・デバイスに関連した統計情報が記述されています。

**Device**

デバイスに関連付けられたスペシャル・ファイルのパス名

**Xmit Pkts/s**

このデバイスによって送信される 1 秒あたりのパケット数

**Xmit Bytes/s**

このデバイスによって送信される 1 秒あたりのバイト数

**Xmit Util**

このデバイスの使用中の時間 (合計時間のパーセント)

#### **Xmit Qlen**

このデバイスで送信されるのを待機中の要求の数 (期間を通じての平均で、現在送信中のトランザクションを含む)

#### **Recv Pkts/s**

このデバイスによって受信される 1 秒あたりのパケット数

#### **Recv Bytes/s**

このデバイスによって受信される 1 秒あたりのバイト数

#### **Recv Demux**

demux レイヤーで費やした時間の合計時間に占める比率

**ネットワーク・デバイス・ドライバー送信統計情報 \_ 宛先別:** 各行には、特定の宛先ホストに関連した送信通信量がデバイス・ドライバー・レベルで記述されます。

各ホストが同一サブネット上にある場合、その宛先ホスト名が表示されます。各ホストが異なるサブネットにある場合、その宛先ホストは ARP プロトコルによって解決され、ブリッジ、ルーター、またはゲートウェイとなる可能性があります。

**Host** 宛先ホスト名。\* (アスタリスク) は、ホスト名が決められないときの送信に使用します。

#### **Pkts/s**

このホストによって送信される 1 秒あたりのパケット数

#### **Xmit Bytes/s**

このホストによって送信される 1 秒あたりのバイト数

### **要約インターネット・ソケット・レポート**

- **オンライン・モード: 各インターネット・プロトコルに関するソケット・コール統計情報 \_ プロセス別:** 各行には、特定プロセスに関連した、このプロトコル・タイプのソケットにおける **read/write** サブルーチン・アクティビティー量が記載されます。詳細オプションを指定しない限り、最大 20 個までのプロセスしかリストされません。レポートの一番下には、このプロトコルのすべてのソケット・コールが合計されています。
- **オフライン・モード: 各プロセスに関するソケット・コール統計情報:** 各行には、特定プロセスに関連したソケットにおける **read/write** サブルーチン・アクティビティー量が記載されます。詳細オプションを指定しない限り、最大 20 個までのプロセスしかリストされません。レポートの一番下には、すべてのソケット・コールが合計されています。

#### **Process**

プロセス名

**PID** プロセス ID 番号

#### **Read Calls/s or Read Ops/s**

このタイプのソケット上で、このプロセスによって 1 秒あたりで呼び出される **read**、**recv**、および **recvfrom** サブルーチンの数。

#### **Read Bytes/s**

上記の呼び出しで要求される 1 秒当たりのバイト数

#### **Write Calls/s or Write Ops/s**

このタイプのソケット上で、このプロセスによって 1 秒あたりで呼び出される **write**、**send**、および **sendto** サブルーチンの数。

## Write Bytes/s

このプロセスによってこのプロトコル・タイプのソケットに書き込まれる秒当たりのバイト数

## 要約 NFS レポート

**各サーバーの NFS クライアント統計情報 \_ ファイル別:** 各行には、このサーバーからリモートにマウントされたファイルに関連した **read/write** サブルーチンのアクティビティー量が記述されます。詳細オプションを指定しない限り、最大 20 個のファイルしかリストされません。レポートの一番下には、このサーバーのすべてのファイルの呼び出しが合計されています。

**File** 単純ファイル名

### Read Calls/s or Read Ops/s

このファイルに関する 1 秒あたりの **read** サブルーチンの数

### Read Bytes/s

上記の呼び出しで要求される 1 秒当たりのバイト数

### Write Calls/s or Write Ops/s

このファイルに関する 1 秒あたりの **write** サブルーチンの数

### Write Bytes/s

このファイルに書き込まれる秒当たりのバイト数

**NFS クライアント RPC 統計情報 \_ サーバー別:** 各行には、特定の NFS サーバーに対してこのクライアントが行う NFS リモート・プロシージャー・コールの数が記述されます。レポートの一番下には、すべてのサーバーの呼び出しが合計されています。

### Server

サーバーのホスト名。(ホスト名が決められない RPC の呼び出しには、\* (アスタリスク) を使用します。)

### Calls/s or Ops/s

このサーバーに対して行われる 1 秒あたりの NFS RPC 呼び出しの数。

**NFS クライアント統計情報 \_ プロセス別:** 各行には、特定プロセスに関連した **read/write** サブルーチンのアクティビティー量が記述されています。詳細オプションを指定しない限り、最大 20 個までのプロセスしかリストされません。レポートの一番下には、すべてのプロセスの呼び出しが合計されています。

### Process

プロセス名

**PID** プロセス ID 番号

### Read Calls/s or Read Ops/s

このプロセスで行われる 1 秒あたりの NFS **read** サブルーチンの数

### Read Bytes/s

上記の呼び出しで要求される 1 秒当たりのバイト数

### Write Calls/s or Write Ops/s

このプロセスで行われる 1 秒あたりの NFS **write** サブルーチンの数

### Write Bytes/s

このプロセスによって NFS マウント・ファイルに書き込まれる 1 秒当たりのバイト数



**NFS サーバー統計情報 \_ クライアント別:** 各行には、特定クライアントのためにこのサーバーが処理する NFS アクティビティ量が記述されています。レポートの一番下には、すべてのクライアントの呼び出しが合計されています。

**Client**

クライアントのホスト名

**Read Calls/s or Read Ops/s**

このクライアントのために処理される 1 秒あたりのリモート読み取り要求の数

**Read Bytes/s**

このクライアントの読み取り呼び出しによって要求される 1 秒当たりのバイト数

**Write Calls/s or Write Ops/s**

このクライアントのために処理される 1 秒あたりのリモート書き込み要求の数

**Write Bytes/s**

このクライアントによって書き込まれる 1 秒当たりのバイト数

**Other Calls/s or Ops/s**

このクライアントのために処理される 1 秒あたりの他のリモート要求の数

## 明細レポート

明細レポートは、指定のレポート・タイプのいずれかで作成されます。これらのレポート・タイプについて、ほとんどの要約レポートの代わりに明細レポートが作成されます。明細レポートには、要約レポートのエントリーごとに 1 つのエントリーが記載され、さらにそのエントリーに関連したトランザクションのタイプごとの統計情報が記載されます。

トランザクション統計情報は、該当タイプのトランザクション数のカウントから構成され、その後に応答時間およびサイズ配布データ (適切であれば) が付けられます。配布データは、標準偏差だけではなく、平均値、最小値、および最大値から構成されます。それらの値のおよそ 3 分の 2 は、average - standard deviation (平均 - 標準偏差) と average + standard deviation (平均 + 標準偏差) との間にあります。サイズはバイト単位で報告されます。応答時間はミリ秒単位で報告されます。

### 2 次レベル割り込みハンドラー CPU 使用統計情報明細 :

**SLIH** 2 次レベル割り込みハンドラーの名前

**Count** このタイプの割り込みの数

**CPU Time (Msec)**

このタイプの割り込み処理のための CPU 使用量統計情報

### ネットワーク・デバイス・ドライバーの詳細統計情報 \_ デバイス別:

**Device**

デバイスに関連付けられたスペシャル・ファイルのパス名

**Recv Packets**

このデバイスによって受信されたパケット数

**Recv Sizes (Bytes)**

受信パケットのサイズ統計情報

**Recv Times (msec)**

受信パケットの処理のための応答時間統計情報

**Xmit Packets**

このホストに送信されたパケット数

**Demux Times (msec)**

demux レイヤーで受信したパケットの処理に関する時間統計情報

**Xmit Sizes (Bytes)**

送信パケットのサイズ統計情報

**Xmit Times (Msec)**

送信パケットの処理のための応答時間統計情報

**ネットワーク・デバイス・ドライバー統計情報明細 \_ ホスト別:**

**Host** 宛先ホスト名

**Xmit Packets**

このデバイスで送信されたパケット数

**Xmit Sizes (Bytes)**

送信パケットのサイズ統計情報

**Xmit Times (Msec)**

送信パケットの処理のための応答時間統計情報

**各インターネット・プロトコルでのソケット・コール統計情報明細 \_ プロセス別: (オンライン・モード)****各プロセスのソケット・コール統計情報明細: (オフライン・モード)****Process**

プロセス名

**PID** プロセス ID 番号

**Reads** このタイプのソケットでこのプロセスによって行われる **read**、**recv**、**recvfrom**、および **recvmsg** サブルーチンの数。

**Read Sizes (Bytes)**

**read** 呼び出しのサイズ統計情報

**Read Times (Msec)**

**read** 呼び出しの応答時間統計情報

**Writes**

このタイプのソケットでこのプロセスによって行われる **write**、**send**、**sendto**、および **sendmsg** サブルーチンの数。

**Write Sizes (Bytes)**

**write** 呼び出しのサイズ統計情報

**Write Times (Msec)**

**write** 呼び出しの応答時間統計情報

**各サーバーの NFS クライアント統計情報明細 \_ ファイル別:**

**File** ファイル・パス名

**Reads** このファイルについての NFS **read** サブルーチンの数

**Read Sizes (Bytes)**

**read** 呼び出しのサイズ統計情報



**Read Times (Msec)**

**read** 呼び出しの応答時間統計情報

**Writes**

このファイルについての NFS **write** サブルーチンの数

**Write Sizes (Bytes)**

**write** 呼び出しのサイズ統計情報

**Write Times (Msec)**

**write** 呼び出しの応答時間統計情報

**NFS クライアント統計情報明細 \_ サーバー別:****Server**

サーバーのホスト名

**Calls** このサーバーに対して行われた NFS クライアントの RPC 呼び出しの数

**Call Times (Msec)**

RPC 呼び出しの応答時間

**NFS クライアント統計情報明細 \_ プロセス別:****Process**

プロセス名

**PID** プロセス ID 番号

**Reads** このプロセスで行われた NFS **read** サブルーチンの数

**Read Sizes (Bytes)**

**read** 呼び出しのサイズ統計情報

**Read Times (Msec)**

**read** 呼び出しの応答時間統計情報

**Writes**

このプロセスで行われた NFS **write** サブルーチンの数

**Write Sizes (Bytes)**

**write** 呼び出しのサイズ統計情報

**Write Times (Msec)**

**write** 呼び出しの応答時間統計情報

**NFS サーバー統計情報明細 \_ クライアント別:****Client**

クライアントのホスト名

**Reads** このクライアントから受信された NFS 読み取り要求の数

**Read Sizes (Bytes)**

読み取り要求のサイズ統計情報

**Read Times (Msec)**

読み取りの応答時間統計情報

**Writes**

このクライアントから受信された NFS 書き込み要求の数

## Write Sizes (Bytes)

書き込み要求のサイズ統計情報

## Write Times (Msec)

書き込み要求の応答時間統計情報

## Other Calls

このクライアントから受信された他の NFS 要求の数

## Other Times (Msec)

他の要求の応答時間統計情報

## 例

1. あるアプリケーション・プログラムの実行時にネットワーク・アクティビティをモニターし、すべてのレポート・タイプを生成するには、以下のように入力します。

```
netpmon  
<run application programs and commands here>  
trcstop
```

**netpmon** コマンドは、自動的にシステム・トレースを開始し、それ自体をバックグラウンドに置きます。この時点で、アプリケーション・プログラムおよびシステム・コマンドを実行できます。 **trcstop** コマンドを発行すると、すべてのレポートが標準出力上に表示されます。

2. CPU および NFS レポート・タイプを生成し、それらのレポートを `nmon.out` ファイルに書き込むには、次のように入力します。

```
netpmon -o nmon.out -0 cpu,nfs  
<run application programs and commands here>  
trcstop
```

**netpmon** コマンドは即座にシステム・トレースを開始します。 **trcstop** コマンドを出すと、入出力のアクティビティ報告は `nmon.out` ファイルに書き込まれます。CPU および NFS のレポートだけが生成されます。

3. すべてのレポート・タイプを生成し、多量の出力を `nmon.out` ファイルに書き込むには、次のように入力します。

```
netpmon -v -o nmon.out  
<run application programs and commands here>  
trcstop
```

多量の出力を指定すると、**netpmon** コマンドはトレースを開始するのに取る手順を示します。要約レポートおよび明細レポートには、20 個のアクティビティ・ファイルおよびプロセスだけでなく、すべてファイルおよびプロセスが含まれます。

4. **netpmon** コマンドをオフライン・モードで使用するには、次のように入力します。

```
trace -a  
run application programs and commands here  
trcsoff  
gennames > gen.out  
trcstop  
trcrpt -r /var/adm/ras/trcfile > tracefile.r  
netpmon -i tracefile.r -n gen.out -o netpmon.out
```

## 関連情報

**trcstop** コマンド、**trace** コマンド、および **gennames** コマンド。

**recv** サブルーチン、**recvfrom** サブルーチン、**send** サブルーチン、**sendto** サブルーチン、および **trcoff** サブルーチン。

---

## netstat コマンド

### 目的

ネットワークの状況を表示します。

### 構文

各プロトコル情報または経路指定テーブル情報に関してアクティブなソケットを表示する

```
/bin/netstat [ -n ] [ { -A -a -o } | { -r -C -i -I Interface } ] [ -f AddressFamily ] [ -p Protocol ] [ Interval ]
```

ネットワーク・データ構造の内容を表示する

```
/bin/netstat [ -m | -M | -s | -ss | -u | -v ] [ -f AddressFamily ] [ -p Protocol ] [ Interval ]
```

仮想インターフェース・テーブルおよびマルチキャスト・フォワード・キャッシュを表示する

```
/bin/netstat -g
```

通信サブシステム全体のパケット数を表示する

```
/bin/netstat -D
```

ネットワーク・バッファ・キャッシュ統計情報を表示する

```
/bin/netstat -c
```

データ・リンク・プロバイダー・インターフェース統計情報を表示する

```
/bin/netstat -P
```

関連統計情報をクリアする

```
/bin/netstat [ -Zc | -Zi | -Zm | -Zs ]
```

### 説明

**netstat** コマンドはアクティブな接続のさまざまなネットワーク関連データ構造の内容を記号で表示します。*Interval* パラメーター (秒) は構成されたネットワーク・インターフェースのパケット・トラフィックに関する情報を表示します。*Interval* パラメーターにはフラグがありません。

### フラグ

- A ソケットと対応するプロトコル制御ブロックのアドレスを示します。このフラグはデフォルト・ディスプレイで機能し、デバッグ用です。
- a すべてのソケットの状態を示します。このフラグを付けなければ、サーバー・プロセスで使われるソケットは示されません。

-c

ネットワーク・バッファ・キャッシュの統計情報を表示します。

ネットワーク・バッファ・キャッシュとは、ネットワークに送信できるデータ・オブジェクトが入れられているネットワーク・バッファのリストです。ネットワーク・バッファ・キャッシュは、データ・オブジェクトが追加または除去されるに従って、動的に大きさが変わります。ネットワーク・バッファ・キャッシュは、ネットワーク入出力に関するパフォーマンスを向上させるために、一部のネットワーク・カーネル・インターフェースによって使用されます。

**netstat -c** コマンドは、次の統計情報を表示します。

```
Network Buffer Cache Statistics:
Current total cache buffer size: 0
Maximum total cache buffer size: 0
Current total cache data size: 0
Maximum total cache data size: 0
Current number of cache: 0
Maximum number of cache: 0
Number of cache with data: 0
Number of searches in cache: 0
Number of cache hit: 0
Number of cache miss: 0
Number of cache newly added: 0
Number of cache updated: 0
Number of cache removed: 0
Number of successful cache accesses: 0
Number of unsuccessful cache accesses: 0
Number of cache validation: 0
Current total cache data size in private segments: 0
Maximum total cache data size in private segments: 0
Current total number of private segments: 0
Maximum total number of private segments: 0
Current number of free private segments: 0
Current total NBC_NAMED_FILE entries: 0
Maximum total NBC_NAMED_FILE entries: 0
```

-C

各経路指定のユーザー構成コストと現行コストを含む、経路指定テーブルを表示します。ユーザー構成コストは、**route** コマンドの **-hopcount** フラグを使用して設定されます。デッド・ゲートウェイ検出によって経路のコストが変更された場合には、現行コストは、ユーザー構成コストと異なる可能性があります。

経路のコストに加えて、各経路に関連する重み情報およびポリシー情報も表示されます。これらのフィールドは、Multipath Routing 機能が使用されている場合にのみ、適用されます。選択可能なポリシー情報は、使用可能な複数の経路の中で現在選択されている経路指定ポリシーを表示します。選択可能なポリシーは次のとおりです。

- デフォルト - 重み付けラウンドロビン (WRR)
- ハッシュ (HSH)
- ランダム (RND)
- 重み付けランダム (WRND)
- 最低使用状況 (LUT)

同一宛先に対して複数の経路が存在する場合 (多重経路)、それらの経路のいずれかには、WRR、HSH、RND、WRND、または LUT のポリシー値が表示されます。このセット内の他のすべての経路には、"- "としてポリシー情報が表示されます。この意味は、このセット内の全経路が最初の経路に表示されたルーティング・ポリシーと同じものを使用していることを示しています。

重み付けフィールドは、重み付けラウンドロビン・ポリシーおよび重み付けランダム・ポリシーで使用される経路に関連する、ユーザー構成の重みです。これらのポリシーの詳細については、**no** コマンドを参照してください。

-D

通信サブシステム内で受信、送信、およびドロップされたパケット数を表示します。

**注:** 統計出力では、フィールド値として N/A が表示されると、カウントが適用できないことを意味します。NFS/RPC 統計情報の場合、RPC を通じて渡される着信パケット数は、NFS を通じて渡されるパケット数と同じなので、これらの数値は NFS/RPC Total フィールドでは合算されないため N/A となります。NFS には、NFS と RPC に特定の発信パケットまたは発信パケット・ドロップ・カウンターはありません。したがって、個々のカウントのフィールド値は N/A であり、累積カウントは NFS/RPC Total フィールドに格納されます。

-f *AddressFamily*

統計情報のレポートまたはアドレス制御ブロックを *AddressFamily* 変数で指定したエントリーに制限します。以下のアドレス・ファミリーが認識されます。

**inet** AF\_INET アドレス・ファミリーを示します。

**inet6** AF\_INET6 アドレス・ファミリーを示します。

**unix** AF\_UNIX アドレス・ファミリーを示します。

-g

仮想インターフェース・テーブルおよびマルチキャスト・フォワード・キャッシュ情報を表示します。-s フラグと組み合わせて使用した場合は、マルチキャスト経路指定情報を表示します。

-i

すべての構成済みインターフェースの状態を示します。インターフェース表示を参照してください。

-l *Interface*

**注:** イーサネット・インターフェースの衝突カウントはサポートされません。*Interface* 変数によって指定された構成済みインターフェースの状態を示します。

-M

ネットワーク・メモリーの MBUF クラスタ・プール統計情報を表示します。

-m

メモリー管理ルーチンが記録した統計情報値を表示します。

<b>-n</b>	ネットワーク・アドレスを番号で表示します。このフラグを指定しない場合、 <b>netstat</b> コマンドは可能なアドレスを解釈し、記号として表示します。このフラグはどのような表示フォーマットでも使用できます。
<b>-o</b>	<b>-a</b> フラグとともに使用され、ソケット・オプション統計情報、フラグ統計情報、およびバッファ統計情報などの、ソケットに関する詳細なデータを表示します。
<b>-p Protocol</b>	<i>Protocol</i> 変数に指定された値に関する統計情報を表示します。Protocol 変数は、プロトコルの定式名またはその別名です。プロトコル名および別名の一部が、 <b>/etc/protocols</b> ファイルにリストされます。null 応答は、報告すべき番号がないことを意味します。静的ルーチンがなければ、 <i>Protocol</i> 変数に指定された値のプログラム報告は認識されません。
<b>-P</b>	Data Link Provider Interface (DLPI) の統計情報を表示します。 <b>netstat -P</b> コマンドは、次のような統計情報を表示します。  DLPI statistics: Number of received packets = 0 Number of transmitted packets = 0 Number of received bytes = 0 Number of transmitted bytes = 0 Number of incoming pkts discard = 0 Number of outgoing pkts discard = 0 Number of times no buffers = 0 Number of successful binds = 0 Number of unknown message types = 0 Status of phys level promisc = 0 Status of sap level promisc = 0 Status of multi level promisc = 0 Number of enab_multi addresses = 0  DLPI がロードされていない場合は、次のように表示されます。  can't find symbol: dl_stats
<b>-r</b>	経路指定テーブルを表示します。 <b>-s</b> フラグと一緒に使用すると、 <b>-r</b> フラグは経路指定の統計情報を表示します。経路指定テーブル表示を参照してください。
<b>-s</b>	各プロトコルの統計情報を表示します。
<b>-ss</b>	ゼロ以外のプロトコル統計情報をすべて簡潔に表示します。
<b>-u</b>	ドメイン・ソケットに関する情報を表示します。
<b>-v</b>	CDLI ベースの通信アダプターに関する統計情報を表示します。このフラグを指定すると、 <b>netstat</b> コマンドが、 <b>entstat</b> 、 <b>tokstat</b> 、および <b>fdistat</b> コマンドに対する統計情報コマンドを実行します。これらのデバイス・ドライバ・コマンドに対して、フラグは発行されません。統計情報出力の説明は、特定のデバイス・ドライバ統計情報コマンドのセクションを参照してください。
<b>-Zc</b>	ネットワーク・バッファ・キャッシュ統計情報をクリアします。
<b>-Zi</b>	インターフェース統計情報をクリアします。
<b>-Zm</b>	ネットワーク・メモリー割り当て統計情報をクリアします。
<b>-Zs</b>	プロトコル統計情報をクリアします。特定のプロトコルの統計情報をクリアする場合は、 <b>-p &lt;protocol&gt;</b> を使用します。例えば、TCP 統計情報をクリアする場合は、 <b>netstat -Zs -p tcp</b> と入力します。

## デフォルト表示

アクティブ・ソケットのデフォルト表示は以下のエントリーを示します。

- ローカル・アドレスとリモート・アドレス
- 送信と受信のキュー・サイズ (バイト単位)
- プロトコル
- プロトコルの内部の状態

ソケットのアドレスが、ネットワークを指定していても、特定のホスト・アドレスを指定していないと、IP アドレスは、 `host.port` あるいは `network.port` というフォーマットになります。ホスト・アドレスは、アドレスを記号ホスト名に解決することができる場合は記号で表示し、ネットワーク・アドレスは `/etc/networks` ファイルに従って記号で表示します。

ホストのシンボル名がわからない場合か、または `-n` フラグを使用した場合は、アドレスはアドレス・ファミリーに従って数値で出力されます。指定されていないアドレスとポートは `*` (アスタリスク) で表示されます。

## インターフェース表示 (`netstat -i`)

インターフェース表示フォーマットは、以下のエントリーについて累積統計情報テーブルを提供します。

- エラー
- 衝突

**注:** イーサネット・インターフェースの衝突カウントはサポートされません。

- 転送されたパケット

インターフェース表示は最大送信単位 (MTU) だけでなく、インターフェース名、番号、アドレスを提供します。

## 経路指定テーブル表示 (`netstat -r`)

経路指定テーブル表示は、使用可能な経路とその状況を示します。各経路は、宛先ホストまたはネットワークと、パケットを転送するときに使うゲートウェイからなっています。

経路は、`A.B.C.D/XX` というフォーマットで提供されます。このフォーマットは 2 つの情報を表します。`A.B.C.D` は宛先アドレスを表し、`XX` は経路に関連付けられたネットマスクを表します。ネットマスクは、設定されたビット数によって表されます。例えば、経路 `9.3.252.192/26` のネットマスクは `255.255.255.192` であり、26 のビットが設定されています。

経路指定テーブルは以下の 10 のフィールドから構成されています。

Flags	経路指定テーブルの <b>flags</b> フィールドは経路の状態を示します。 <b>A</b> アクティブ・デッド・ゲートウェイ検出がこの経路で使用可能になっています。このフィールドは、AIX 5.1 以降にだけ適用されます。 <b>U</b> 立ち上がっています。 <b>H</b> 経路はネットワークあてではなくホスト宛です。 <b>G</b> 経路はゲートウェイ宛です。 <b>D</b> 経路はリダイレクトによって動的に作成されています。 <b>M</b> 経路はリダイレクトによって修正されています。 <b>L</b> リンク・レベルのアドレスは経路エントリーにあります。 <b>c</b> この経路へのアクセスは、クローン経路を作成します。 <b>W</b> この経路はクローン経路です。 <b>1</b> プロトコル特定の経路指定フラグ #1。 <b>2</b> プロトコル特定の経路指定フラグ #2。 <b>3</b> プロトコル特定の経路指定フラグ #3。 <b>b</b> 経路はブロードキャスト・アドレスを示します。 <b>e</b> 割り当てキャッシュ・エントリーを使用します。 <b>l</b> 経路はローカル・アドレスを示します。 <b>m</b> 経路はマルチキャスト・アドレスを示します。 <b>P</b> ピン固定経路。 <b>R</b> 到達不可能なホストまたはネットワーク。 <b>S</b> 手動で追加。 <b>u</b> 使用可能な経路。 <b>s</b> グループ経路指定停止検索オプションがこの経路で使用可能になっています。
Gateway	ローカル・ホストに接続されたインターフェースごとに直接経路が作成され ます。 これらのエントリーの <b>gateway</b> フィールドは出力インターフェースのアドレス を示します。
Refs	経路の現在有効な使用数を示します。 接続指向のプロトコルでは接続中単一の経 路を確保し続けますが、コネクションレス型プロトコルは同じ宛先に送信する間 経路を確保します。
Use	この経路を介して送信したパケット数を示します。
PMTU	パス最大転送ユニット (PMTU) を示します。 AIX 5.3 は PMTU 列を表示しませ ん。
Interface	経路に使用したネットワーク・インターフェースを示します。
Exp	経路満了までに残っている時間 (分単位) を表示します。
Groups	その経路と関連したグループ ID のリストを提供します。
Netmasks	システムに適用されるネットマスクをリストします。



Route Tree for  
Protocol Family

既存の経路に対する有効なアドレス・ファミリーを指定します。このフィールドにサポートされる値は以下の通りです。

- 1 UNIX アドレス・ファミリーを指定します。
- 2 IP アドレス・ファミリーを指定します。(例えば、TCP および UDP)

他のアドレス・ファミリーの詳細は、`/usr/include/sys/socket.h` ファイルを参照してください。

*Interval* パラメーターに値を指定すると、**netstat** コマンドはネットワーク・インターフェースに関連する統計情報の実行中カウントを表示します。第 1 の欄は 1 次インターフェース (自動構成中に見つかった第 1 のインターフェース) を表し、第 2 の欄はすべてのインターフェースに関する情報の要約を表します。

**-i** フラグを使うと、1 次インターフェースを別のインターフェースと置換できます。各情報画面の先頭行には、システムが最後に再開されてから累積された統計情報の要約が入っています。それ以後の出力行は、指定された長さの間隔に渡って累積された値を表示します。

## 例

1. インターネット・インターフェースの経路指定テーブルの情報を表示するには、以下のように入力します。

```
netstat -r -f inet
```

以下の出力が作成されます。

```
Routing tables
Destination Gateway      Flags Refs Use  PMTU If  Exp Groups Netmasks:
(root node)
(0)0 ffff f000 0
(0)0 ffff f000 0
(0)0 8123 262f 0 0 0 0 0
(root node)
```

```
Route Tree for Protocol Family 2:
(root node)
default      129.35.38.47  UG   0 564 -  tr0  -
loopback     127.0.0.1    UH   1 202 -  lo0  -
129.35.32    129.35.41.172 U    4 30  -  tr0  -  +staff
129.35.32.117 129.35.41.172 UGHW 0 13 1492 tr0 30
192.100.61   192.100.61.11 U    1 195 -  en0  -
(root node)
```

```
Route Tree for Protocol Family 6:
(root node)
(root node)
```

`-r -f inet` フラグはすべての構成済みインターネット・インターフェースの経路指定テーブル情報の要求を示します。ネットワーク・インターフェースは **Interface** 欄にリストされます。`en` は標準イーサネット・インターフェース、`tr` はトークンリング・インターフェースを示します。ゲートウェイ・アドレスはドット 10 進数フォーマットです。

**注:** AIX 5.3 は PMTU 列を表示しません。

2. GRE プロトコルの統計情報を表示するには、以下のように入力します。

```
netstat -s -p gre
```

以下の出力が作成されます。

```

GRE Interface gre0
  10 number of times gre_input got called
  8 number of times gre_output got called
  0 packets received with protocol not supported
  0 packets received with checksum on
  0 packets received with routing present
  0 packets received with key present
  0 packets received with sequence number present
  0 packets received with strict source route present
  0 packets received with recursion control present
  0 packets received where reserved0 non-zero
  0 packets received where version non-zero
  0 packets discarded
  0 packets dropped due to network down
  0 packets dropped due to protocol not supported
  0 packets dropped due to error in ip output routine
  0 packets got by NAT
  0 packets got by NAT but not TCP packet
  0 packets got by NAT but with IP options

```

3. インターネット・インターフェースのインターフェース情報を表示するには、以下のように入力します。

```
netstat -i -f inet
```

AIX 4.2 を使用している場合は、次の出力が作成されます。

Name	Mtu	Network	Address	Ipkts	Ierrs	Opkts	Oerrs	Coll
lo0	1536	<Link>		4	0	4	0	0
lo0	1536	127	loopback	4	0	4	0	0
en0	1500	<Link>		96	0	67	0	0
en0	1500	192.100.61	nullarbor	96	0	67	0	0
tr0	1500	<Link>		44802	0	11134	0	0
tr0	1500	129.35.32	stnullarb	44802	0	11134	0	0

AIX 4.3 を使用している場合は、次の出力が作成されます。

Name	Mtu	Network	Address	Ipkts	Ierrs	Opkts	Oerrs	Coll
lo0	16896	Link#1		5161	0	5193	0	0
lo0	16896	127	localhost	5161	0	5193	0	0
lo0	16896	:::1		5161	0	5193	0	0
en1	1500	Link#2	8.0.38.22.8.34	221240	0	100284	0	0
en1	1500	129.183.64	infoserv.frec.bul	221240	0	100284	0	0

-i -f inet フラグは、すべての構成済みインターネット・インターフェースの状況に対する要求を示します。ネットワーク・インターフェースは Name 欄にリストされています。lo はループバック・インターフェース、en は標準イーサネット・インターフェース、tr はトークンリング・インターフェースを示します。

4. 各プロトコルの統計情報を表示するには、以下のように入力します。

```
netstat -s -f inet
```

以下の出力が作成されます。

```

ip:
:
  44485 total packets received
  0 bad header checksums
  0 with size smaller than minimum
  0 with data size < data length
  0 with header length < data size
  0 with data length < header length
  0 with bad options
  0 with incorrect version number
  0 fragments received
  0 fragments dropped (dup or out of space)

```

```

0 fragments dropped after timeout
0 packets reassembled ok
44485 packets for this host
0 packets for unknown/unsupported protocol
0 packets forwarded
0 packets not forwardable
0 redirects sent
1506 packets sent from this host
0 packets sent with fabricated ip header
0 output packets dropped due to no bufs, etc.
0 output packets discarded due to no route
0 output datagrams fragmented
0 fragments created
0 datagrams that can't be fragmented
0 IP Multicast packets dropped due to no receiver
0 successful path MTU discovery cycles
0 path MTU rediscovery cycles attempted
0 path MTU discovery no-response estimates
0 path MTU discovery response timeouts
0 path MTU discovery decreases detected
0 path MTU discovery packets sent
0 path MTU discovery memory allocation failures
0 ipintrq overflows

icmp:
0 calls to icmp_error
0 errors not generated 'cuz old message was icmp
Output histogram:
  echo reply: 6
0 messages with bad code fields
0 messages < minimum length
0 bad checksums
0 messages with bad length
Input histogram:
  echo: 19
6 message responses generated

igmp:defect
0 messages received
0 messages received with too few bytes
0 messages received with bad checksum
0 membership queries received
0 membership queries received with invalid field(s)
0 membership reports received
0 membership reports received with invalid field(s)
0 membership reports received for groups to which we belong
0 membership reports sent

tcp:
1393 packets sent
  857 data packets (135315 bytes)
  0 data packets (0 bytes) retransmitted
  367 URG only packets
  0 URG only packets
  0 window probe packets
  0 window update packets
  170 control packets
1580 packets received
  790 acks (for 135491 bytes)
  60 duplicate acks
  0 acks for unsent data
  638 packets (2064 bytes) received in-sequence
  0 completely duplicate packets (0 bytes)
  0 packets with some dup. data (0 bytes duped)
  117 out-of-order packets (0 bytes)
  0 packets (0 bytes) of data after window
  0 window probes

```

```

    60 window update packets
    0 packets received after close
    0 discarded for bad checksums
    0 discarded for bad header offset fields
0 connection request
58 connection requests
61 connection accepts
118 connections established (including accepts)
121 connections closed (including 0 drops)
0 embryonic connections dropped
845 segments updated rtt (of 847 attempts)
0 resends due to path MTU discovery
0 path MTU discovery terminations due to retransmits
0 retransmit timeouts
    0 connections dropped by rexmit timeout
0 persist timeouts
0 keepalive timeouts
    0 keepalive probes sent
    0 connections dropped by keepalive

```

```

udp:
  42886 datagrams received
:
  0 incomplete headers
  0 bad data length fields
  0 bad checksums
  0 dropped due to no socket
  42860 broadcast/multicast datagrams dropped due to no

```

```

socket
  0 socket buffer overflows
  26 delivered
  106 datagrams output

```

ip はインターネット・プロトコルを指定します。icmp は情報制御メッセージ・プロトコルを、tcp は伝送制御プロトコルを、udp はユーザー・データグラム・プロトコルを指定します。

**注:** AIX 5.3 は、IP プロトコルの PMTU 統計情報を表示しません。

5. デバイス・ドライバーの統計情報を表示するには、以下のように入力します。

```
netstat -v
```

netstat -v コマンドは、実行中の CDLI ベースのデバイス・ドライバーごとに、統計情報を表示します。このコマンドのサンプル出力については、**tokstat** コマンド、**entstat** コマンド、または **fdistat** コマンドのセクションを参照してください。

6. マルチキャストが使用可能になっているインターフェースに関する情報と、グループ・メンバーシップを表示するには、以下のように入力します。

```
netstat -a -I interface
```

例えば、802.3 インターフェースを指定すると、以下の出力が生成されます。

Name	Mtu	Network	Address	Ipkts	Ierrs	Opkts	Oerrs	Coll
et0	1492	<Link>		0	0	2	0	0
et0	1492	9.4.37	hun-eth 224.0.0.1 02:60:8c:0a:02:e7 01:00:5e:00:00:01	0	0	2	0	0

**-I interface** の代わりにフラグ **-i** を指定すると、構成済みのすべてのインターフェースがリストされま  
す。ネットワーク・インターフェースは Name 欄にリストされます。**lo** はループバック・インターフ

エースを示し、 **et** は IEEE 802.3 インターフェースを示し、 **tr** はトークンリング・インターフェースを示し、 **fi** は FDDI インターフェースを示します。

アドレス欄の意味は以下のとおりです。各インターフェースのシンボル名が表示されます。このシンボル名の下に、そのインターフェース上で結合されているマルチキャスト・グループのグループ・アドレスが表示されます。グループ・アドレス 224.0.0.1 は、すべてのマルチキャスト・インターフェースが属する特殊な *all-hosts-group* です。インターフェースの MAC アドレス (コロン表記) はグループ・アドレスの後に表示され、IP マルチキャストの代わりに特定のインターフェース向けに使用可能となっている他の MAC レベル・アドレスのリストも表示されます。

7. 通信サブシステム内のパケット・カウントを表示するには、以下のように入力します。

```
netstat -D
```

以下の出力が生成されます。

Source	Ipkts	Opkts	Idrops	Odrops
tok_dev0	720	542	0	0
ent_dev0	114	4	0	0
-----				
Devices Total	834	546	0	0
-----				
tok_dd0	720	542	0	0
ent_dd0	114	4	0	0
-----				
Drivers Total	834	546	0	0
-----				
tok_dmx0	720	N/A	0	N/A
ent_dmx0	114	N/A	0	N/A
-----				
Demuxer Total	834	N/A	0	N/A
-----				
IP	773	767	0	0
TCP	536	399	0	0
UDP	229	93	0	0
-----				
Protocols Total	1538	1259	0	0
-----				
lo_if0	69	69	0	0
en_if0	22	8	0	0
tr_if0	704	543	0	1
-----				
Net IF Total	795	620	0	1
-----				
NFS/RPC Client	519	N/A	0	N/A
NFS/RPC Server	0	N/A	0	N/A
NFS Client	519	N/A	0	N/A
NFS Server	0	N/A	0	N/A
-----				
NFS/RPC Total	N/A	519	0	0

(Note: N/A -> Not Applicable)

8. アクティブなソケットの詳細データを表示するには、以下のように入力します。

```
netstat -aon
```

以下のような出力が表示されます。

```
Active Internet connections (including servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         (state)
tcp4      0      0 *.13                    *.*                      LISTEN
    so_options: (ACCEPTCONN|REUSEADDR)
    q0len:0 qlen:0 qlimit:1000    so_state: (PRIV)
    timeo:0 uid:0
```

```

so_special: (LOCKBALE|MEMCOMPRESS|DISABLE)
so_special2: (PROC)
sndbuf:
    hiwat:16384 lowat:4096 mbcnt:0 mbmax:65536
rcvbuf:
    hiwat:16384 lowat:1 mbcnt:0 mbmax:65536
    sb_flags: (SEL)
TCP:
    mss:512

tcp      0      0 *.21          *.*          LISTEN

so_options: (ACCEPTCONN|REUSEADDR)
q0Ten:0 qlen:0 qlimit:1000      so_state: (PRIV)
timeo:0 uid:0
so_special: (LOCKBALE|MEMCOMPRESS|DISABLE)
so_special2: (PROC)
sndbuf:
    hiwat:16384 lowat:4096 mbcnt:0 mbmax:65536
rcvbuf:
    hiwat:16384 lowat:1 mbcnt:0 mbmax:65536
    sb_flags: (SEL)
TCP:
    mss:512

```

```

.....
.....

```

9. 経路指定テーブルを表示するには、次のように入力します。

```
netstat -rn
```

以下のような出力が表示されます。

```

Routing tables
Destination      Gateway          Flags  Refs    Use  If  PMTU  Exp  Groups
-----
Route Tree for Protocol Family 2 (Internet):
default          9.3.149.65      UG      0       24  en0  -   -
9.3.149.64       9.3.149.88      UHSb    0        0  en0  -   -      =>
9.3.149.64/27    9.3.149.88      U        1        0  en0  -   -
9.3.149.88       127.0.0.1       UGHS    0        1  lo0  -   -
9.3.149.95       9.3.149.88      UHSb    0        0  en0  -   -
127/8            127.0.0.1       U       11       174  lo0  -   -

Route Tree for Protocol Family 24 (Internet v6):
::1              ::1             UH      0        0  lo0  -   -

```

注: AIX 5.3 は PMTU 列を表示しません。

行の末尾の文字 => は、その行が次の行の経路の重複経路であることを示しています。

ループバック経路 (9.3.149.88、127.0.0.1) とブロードキャスト経路 (ブロードキャストを示す b を含むフラグ・フィールド付き) は、インターフェースの構成時に自動的に作成されます。ブロードキャスト経路は 2 つ、すなわちサブネット・アドレスとサブネットのブロードキャスト・アドレスにそれぞれ 1 つずつ追加されます。ループバック経路とブロードキャスト経路が存在すると、パフォーマンスが向上します。

## 関連情報

**atmstat** コマンド、**entstat** コマンド、**fddistat** コマンド、**iostat** コマンド、**no** コマンド、**tokstat** コマンド、**trpt** コマンド、**vmstat** コマンド。

**hosts** ファイル・フォーマット、**networks** ファイル・フォーマット、**protocols** ファイル・フォーマット、**services** ファイル・フォーマット。

パフォーマンス・マネージメントの ネットワーク・パフォーマンスのセクション。

ネットワークおよびコミュニケーションのマネージの TCP/IP 経路指定ゲートウェイ、命名、TCP/IP アドレスリング、TCP/IP のネットワーク・インターフェース、TCP/IP のプロトコル、および TCP/IP の経路指定のセクション。

---

## newaliases コマンド

### 目的

メール別名ファイルから別名データベースの新規のコピーを作成します。

### 構文

**newaliases**

### 説明

**newaliases** コマンドは、**/etc/aliases** ファイルから別名データベースの新しいコピーを作成します。このファイルを変更したときは、その都度コマンドを実行して変更を有効にする必要があります。このコマンドを実行することは、**-bi** フラグを指定して **sendmail** コマンドを実行するのと同じです。

### 終了状況

0                   正常に終了しました。  
>0                   エラーが発生しました。

### ファイル

**/usr/sbin/newaliases**  
**/etc/mailaliases**  
**/etc/aliasesDB** directory

**newaliases** コマンドが入っています。  
メール別名ファイル・コマンドのソースが入っています。  
**newaliases** コマンドによって作成されたバイナリー・ファイルが入っています。

### 関連情報

**sendmail** コマンド。

ネットワークおよびコミュニケーションのマネージのメール別名 および 別名データベースの構築のセクション。

---

## newform コマンド

### 目的

テキスト・ファイルのフォーマットを変更します。

## 構文

```
newform [ -s ] [ -f ] [ -a [ Number ] ] [ -b [ Number ] ] [ -c [ Character ] ] [ -e [ Number ] ]  
[ -i [ TabSpec ] ] [ -l [ Number ] ] [ -o [ TabSpec ] ] [ -p [ Number ] ] [ File ... ]
```

## 説明

**newform** コマンドは *File* パラメーターで指定したファイルから行を読み取り (デフォルトは標準入力)、フォーマットされた行を標準出力に書き出します。有効なコマンド行フラグに従って行が再フォーマットされます。

**-s** フラグを除き、コマンド行フラグは任意の順序で入力でき、繰り返すことも、*File* パラメーターと一緒に指定することもできます。ただし、システムはコマンド行フラグを指定された順序で処理します。例えば、**-c** フラグは **-a** フラグと **-p** フラグの動作を修正するので、修正したい **-p** フラグや **-a** フラグの前に **-c** フラグを指定します。また **-l** (L の小文字) フラグは、**-a**、**-b**、**-e**、および **-p** フラグの動作を修正するので、修正したいフラグの前に **-l** フラグを指定します。例えば、**-e15 -l60** というフラグ順序と **-l60 -e15** というフラグ順序では、結果が異なります。フラグはコマンド行に指定したすべてのファイルに適用されます。

終了値が 0 の場合は、正常終了を示します。終了値 1 の場合は、エラーを示します。

### 注:

1. **newform** コマンドは通常、物理的な文字を記録します。ただし **-i** および **-o** フラグの場合、**newform** コマンドは適切な論理カラムでタブを合わせるためバックスペースも記録します。
2. **newform** コマンドは、(**-i-** または **-o-** フラグを使用して) *TabSpec* 変数を標準入力から読み込んだ場合はプロンプトを出しません。
3. **-f** フラグが指定され、かつ最後に指定した **-o** フラグが **-o-** であり、**-o-** または **-i-** が先行する場合、このタブ指定フォーマット行は正しくありません。
4. **-p**、**-l**、**-e**、**-a**、または **-b** のフラグに指定した値が 1 よりも大きい有効な 10 進数でない場合、指定した値は無視され、デフォルトの動作が行われます。

## フラグ

**-a** [ *Number* ] 行の長さが有効な行長より短い場合、指定された数の文字を行の終わりに追加します。数値を指定しないと、**-a** フラグのデフォルトは 0 となり、有効な行長にするのに必要な文字数を追加します。**-c** [ *Character* ] および **-p** [ *Number* ] フラグも参照してください。

**-b** [ *Number* ] 行の長さが有効な行長より長い場合、行の始めから指定された数の文字を削除します。行に *Number* パラメーターで指定した数より少ない文字しか入っていないと、行全体が削除され、そこにブランク行が表示されます。**-l** [ *Number* ] フラグも参照してください。*Number* 変数を指定せず、**-b** フラグを指定した場合は、デフォルトの動作により、有効な行長にするために必要な文字数が切り捨てられます。

このフラグは、次のように COBOL プログラムからシーケンス番号を削除するために使用できます。

```
newform -l1-b7 file-name
```

**-b** フラグが活動化するように **-l1** フラグを使って有効な行長をファイル内の既存の行よりも短く設定しなければなりません。

**-c** [ *Character* ] 接頭部/追加文字を *Character* 変数で指定した文字に変更します。デフォルトの文字はスペースで、**-a** フラグおよび **-p** フラグの前に指定されているときに有効です。



- e** [ *Number* ] 行末から指定した数だけの文字を切り捨てます。それ以外は、**-b** [ *Number* ] フラグと同じです。
- f** 他の行を書き込む前にタブ指定フォーマット行を標準出力に書き出します。表示されたタブ指定フォーマット行は、最後の **-o** フラグで指定されたフォーマットに対応します。**-o** フラグを指定しない場合は、表示される行はデフォルト指定 **-8** を含みません。
- i** [ *TabSpec* ] 入力のすべてのタブを *TabSpec* 変数で指定した数のスペースで置換します。
- この変数は **tabs** コマンドで記述したすべてのタブ指定フォーマットを認識します。
- TabSpec* 変数の値として **-** (マイナス符号) を指定すると、**newform** コマンドは、標準入力から読み取られた最初の行にタブ指定があるものと見なします。*TabSpec* のデフォルト値は **-8** です。*TabSpec* の値が **-0** の場合は、タブはないものと見なします。ある場合は、値が **-1** として扱います。
- l** [ *Number* ] 有効行長を指定文字数に設定します。*Number* 変数を指定しない場合、**-l** フラグのデフォルトの **72** が使用されます。**-l** フラグを指定しない場合のデフォルトの行長は **80** 文字です。タブおよびバックスペースは、**1** 文字と見なされることに注意してください (**-i** フラグを使用すると、タブはスペースに置換されます)。**-l** フラグは、**-b** および **-e** フラグの前に指定しなければなりません。
- o** [ *TabSpec* ] タブ指定に従って入力のスペースを出力でタブに置換します。*TabSpec* のデフォルト値は **-8** です。*TabSpec* の値が **-0** の場合は、スペースを出力でタブに変換しないことを意味します。
- p** [ *Number* ] 行の長さが有効な行長より短い場合、指定された数の文字を行の始めに追加します。デフォルトでは有効な行長にするために必要な文字数を付加します。**-c** フラグも参照してください。
- s** 各行の最初のタブまでの先行する文字を除去し、除去した文字の最大 **8** 文字を行末に追加します。**8** 文字を超えて除去した場合は (最初のタブを除き)、**8** 番目の文字は **\*** (アスタリスク) と置き換えられ、その右側の文字は捨てられます。最初のタブは常に破棄されます。
- 除去した文字は、他の指定フラグがその行に適用されるまで内部的に保存されます。文字は次に処理した行の終わりに追加されます。

注: **-a**、**-b**、**-e**、**-l** (L の小文字)、および **-p** フラグの値は、**LINE\_MAX** もしくは **2048** バイトを超えてはなりません。

## 例

次のフォーマットのファイルを変換します。

- 先行する数字がある
- 1 つ以上のタブがある
- 各行にテキストが入っている

次のように変換します。

- テキストで始まり、最初のタブ以外のタブをスペースに変換する
- 72 桁までスペースを埋め込む (あるいは 72 桁まで切り捨てる)
- 先行する数字は 73 桁から始める

次のように入力します。

```
newform -s -i -l -a -e filename
```

行ごとに、タブがないファイルで **-s** フラグを使用すると、**newform** コマンドは次のエラー・メッセージを表示して停止します。

```
newform: 0653-457 The file is not in a format supported by the -s flag.
```

## 関連情報

**tabs** コマンド、**csplit** コマンド。

---

## newgrp コマンド

### 目的

1 次グループの ID を変更します。

### 構文

```
newgrp [ - ] [ -l ] [ Group]
```

### 説明

**newgrp** コマンドは 1 次グループの ID を変更します。このコマンドを実行すると、新しいシェルに入り、実グループの名前が *Group* パラメーターで指定した名前に変更されます。デフォルトでは、**newgrp** コマンドは実グループを **/etc/passwd** ファイル内の指定したグループに変更します。

**注:** **newgrp** コマンドは、標準入力からの入力を使用しません。また、スクリプト内からは実行できません。

**newgrp** コマンドはグループ ID 番号ではなく、グループ名のみを認識します。変更結果は現在のセッション中でのみ有効です。実グループ名は、既にメンバーになっているグループ名にしか変更できません。root ユーザーであれば、メンバーになっているかどうかに関係なく、実グループを任意のグループに変更できます。

**注:** **newgrp** コマンドを実行すると、常にシェルが新しいシェルに置き換えられます。このコマンドは、正常に実行されるかどうかに関係なくシェルを置き換えます。したがって、このコマンドはエラー・コードを戻しません。

### フラグ

- 環境を新グループのログイン環境に変更します。
- l - フラグと同じ値を示します。

### セキュリティ

アクセス制御：このコマンドはトラステッド・コンピューティング・ベース (TCB) 内のプログラムとしてインストールしなければなりません。このコマンドは、**setuid** (SUID) ビットが設定されている root ユーザーが所有しなければなりません。

### 終了状況

**newgrp** コマンドが新しいシェル実行環境を正常に作成すると、グループ ID が正常に変更されたかどうかに関係なく、終了状況は現行シェルの状況となります。それ以外の場合は、次の終了値が戻されます。

>0 エラーが発生しました。

## 例

1. 現行シェル・セッションの実グループ ID を `admin` に変更するには、次のように入力します。

```
newgrp admin
```

2. 実グループ ID を元のログイン・グループに変更するには、次のように入力します。

```
newgrp
```

## ファイル

`/etc/group`

グループ・ファイルを示します。グループ ID が入っています。

`/etc/passwd`

パスワード・ファイルを示します。ユーザー ID が入っています。

## 関連情報

`login` コマンド、`setgroups` コマンド。

---

## newkey コマンド

### 目的

`/etc/publickey` ファイル内に新しいキーを作成します。

### 構文

```
/usr/sbin/newkey [ -h HostName] [ -u UserName]
```

### 説明

`newkey` コマンドは `/etc/publickey` ファイルに新しいキーを作成します。これは通常、ネットワーク管理者が、ネットワーク上のユーザーおよび `root` ユーザーのための公開鍵を設定するために、`Network Information Services (NIS)` マスター・マシン上で実行するコマンドです。これらのキーは、セキュア・リモート・プロシーチャー・コール (RPC) プロトコルまたはセキュア・ネットワーク・ファイルシステム (NFS) を使用するために必要です。

`newkey` コマンドは `UserName` パラメーターで指定されたユーザーのログイン・パスワードを入力するようにプロンプトを表示します。次に、新しいキーのペアを `/etc/publickey` ファイルに作成し、`publickey` データベースを更新します。キーのペアはユーザーの公開鍵と秘密鍵から構成され、指定ユーザーのログイン・パスワードとともに暗号化されます。

必ずしもこのプログラムを使う必要はありません。`chkey` コマンドを使うとユーザー独自のキーを作成できます。

Web-based System Manager (wsm) のネットワーク・アプリケーションを使用してネットワーク特性を変更できます。このコマンドは、System Management Interface Tool (SMIT) `smit newkey` 高速パスを使用して実行できます。

## フラグ

- h** *HostName* *HostName* パラメーターで指定されたコンピューターで、root ユーザー用の新しい公開鍵を作成します。このパラメーターの root パスワードを入力するように求めるプロンプトを表示します。
- u** *UserName* *UserName* パラメーターで指定されたユーザー用の新しい公開鍵を作成します。このパラメーターの NIS パスワードを入力するように求めるプロンプトを表示します。

## 例

1. ユーザー用の新しい公開鍵を作成するには、次のように入力します。

```
newkey -u john
```

この例では、**newkey** コマンドは john というユーザー用の新しい公開鍵を作成します。

2. ホスト zeus 上の root ユーザーの新しい公開鍵を作成するには、次のように入力します。

```
newkey -h zeus
```

この例では、**newkey** コマンドは zeus という名前のホスト上で root ユーザーの新しい公開鍵を作成します。

## ファイル

**/etc/publickey** ユーザー用に暗号化されたキーを保存します。

## 関連情報

**chkey** コマンド、**keylogin** コマンド。

**keyserv** デーモン。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

オペレーティング・システムおよびデバイスのマネージの System Management Interface Toolのセクション。

ネットワークおよびコミュニケーションのマネージのネットワーク・ファイルシステム (NFS) の概要のセクション。

セキュリティーのセキュア NFS を使用したファイルシステムのエクスポートのセクション、セキュア NFS を使用したファイルシステムのマウントのセクション。

*AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*の Network Information Service (NIS) のセクション。

NIS のリファレンス。

---

## news コマンド

### 目的

システム・ニュース・エントリーを標準出力に書き出します。

### 構文

`news [ -a | -n | -s | Item ...]`

### 説明

`news` コマンドは、システム・ニュース・エントリーを標準出力に書き出します。このコマンドは、システムに関するニュースを常にユーザーに通知します。各ニュース・エントリーは `/var/news` ディレクトリーの別のファイルに入っています。通常、ユーザーは、`$HOME/.profile` ファイルまたはシステムの `/etc/profile` ファイルに、`-n` フラグを付けた `news` コマンドを追加しておいて、ログイン時に毎回実行するようにします。このディレクトリーに書き込み許可を持つユーザーは、ニュース・エントリーを作成できます。読み取り許可がなくても、ニュース・エントリーを作成できます。

フラグを指定しないで `news` コマンドを実行する場合、`/var/news` ファイルのすべての現行ファイルを最新のものから表示します。このコマンドで `-a` フラグを指定すると、すべてのニュース・エントリーが表示されます。`-n` フラグを指定すると、読んでいないニュース・エントリーの名前のみが表示されます。`-s` フラグを使用すると、未読のニュース・エントリーの数が表示されます。また、`Item` パラメーターを使用すると、表示したいファイルを指定できます。

各ファイルの前に適切なヘッダーが付きます。古いニュースの通知を避けるため、`news` コマンドは現在の時刻を保管します。`news` コマンドは現在時刻が `$HOME/.news_time` ファイルの最終変更日付であると見なします。ニュースを読むたびに、このファイルの修正時間は読んだ時間に変更されます。この後に通知されたニュース・エントリーだけが現行ファイルと見なされます。

ニュース・エントリーが表示されているときに割り込み (Ctrl-C) キー・シーケンスを押すと、そのエントリーの表示は停止し、次のエントリーが表示されます。Ctrl-C キー・シーケンスを再度押すと、`news` コマンドは終了します。

注: ニュース・エントリーはマルチバイト文字を含むことができます。

### フラグ

- `-a` 現在の時間に関係なくニュース・エントリーをすべて表示します。現在の時間は変わりません。
- `-n` 内容を表示せずに現行ニュース・エントリー名を報告します。現在の時間は変わりません。
- `-s` 名前や内容を表示せずに現行ニュース・エントリー数を報告します。現在の時間は変わりません。

### 例

1. ニュースを最後に読み取って以降通知されたエントリーを表示するには、次のように入力します。

```
news
```

2. すべてのニュース・エントリーを表示するには、次のように入力します。

```
news -a | pg
```

すべてのニュース・エントリーは、既に読まれたことがあるかどうかにかかわらず、一度に 1 ページずつ (1 pg) 表示されます。

3. 未読のニュース・エントリーの名前を表示するには、次のように入力します。

```
news -n
```

各名前は **/var/news** ディレクトリーのファイルです。

4. 特定のニュース・エントリーを表示するには、次のように入力します。

```
news newusers services
```

このコマンド・シーケンスは **newusers** と **services** に関するニュースを表示します。これらの名前は、**news -n** コマンドによってリストされています。

5. 未読のニュース・エントリーの数を表示するには、次のように入力します。

```
news -s
```

6. 誰でも読めるようにニュースを通知するには、次のように入力します。

```
cp schedule /var/news
```

これは、**schedule** ファイルを **/var/news** ディレクトリーにコピーし、**/var/news/schedule** ファイルを作成します。これには、**/var/news** ディレクトリーに対する書き込み許可が必要です。

## ファイル

<b>/usr/bin/news</b>	<b>news</b> コマンドが入っています。
<b>/etc/profile</b>	システム・プロファイルが入っています。
<b>/var/news</b>	システム・ニュース・エントリー・ファイルが入っています。
<b>\$HOME/.news_time</b>	<b>news</b> コマンドを最後に呼び出した日付を示します。

## 関連情報

**pg** コマンド。

**/etc/security/environ** ファイル、**profile** ファイル。

---

## next コマンド

### 目的

次のメッセージを表示します。

### 構文

```
next [ +Folder ] [ -header | -noheader ] [ -showproc CommandString | -noshowproc ]
```

### 説明

**next** コマンドはシステムがメッセージ・ハンドラー (MH) フォルダの次のメッセージ・フィールドに割り当てる番号を表示します。**next** コマンドは、**next** 値をメッセージとして指定した **show** コマンドと同じです。

**next** コマンドは **show** プログラムにリンクし、スイッチを **showproc** プログラムに渡します。 **next** 値にリンクし、そのリンクに **next** 以外の名前を付ける場合、リンクは **next** コマンドではなく、**show** コマンドと同様に機能します。

**show** コマンドは認識されないフラグを、リストを行うプログラムに渡します。 **next** コマンドは、いくつかのフラグをリストするプログラムに渡します。

## フラグ

<b>+Folder</b>	表示したいメッセージが入っているフォルダーを指定します。
<b>-header</b>	表示するメッセージの記述を 1 行で表示します。記述にはフォルダー名とメッセージ番号が含まれます。これはデフォルトです。
<b>-help</b>	コマンド構文、使用可能なスイッチ (トグル)、およびバージョン情報をリストします。
<b>-noheader</b>	<b>注:</b> MH の場合は、このフラグ名を省略することはできません。
<b>-noshowproc</b>	表示する各メッセージを 1 行での記述で表示するのを抑制します。
<b>-showproc</b> <i>CommandString</i>	リスト作成を行うため <b>/usr/bin/cat</b> ファイルを使用します。これはデフォルトです。 指定されたコマンド・ストリングを使用してリスト作成を実行します。

## 例

1. 現行フォルダー内の次のメッセージを表示するには、以下のように入力します。

```
next
```

システムは次のようなメッセージで応答します。

```
(Message schedule: 10)
```

メッセージのテキストも表示されます。この例では、現行フォルダー `schedule` のメッセージ 10 が次のメッセージです。

2. `project` フォルダー内の次のメッセージを表示するには、以下のように入力します。

```
next +project
```

システムはメッセージのテキストと以下のようなヘッダーを表示して応答します。

```
(Message project: 5)
```

## ファイル

<b>\$HOME/.mh_profile</b>	ユーザーの MH プロファイルを指定します。
<b>/usr/bin/next</b>	<b>next</b> コマンドが入っています。

## 関連情報

**prev** コマンド、**show** コマンド。

**.mh\_alias** ファイル・フォーマット、**.mh\_profile** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ のメール・アプリケーションのセクション。



---

## nfs.clean コマンド

### 目的

NFS および NIS 操作を停止します。

### 構文

```
/etc/nfs.clean [-d][-y][-t nfs|nis]
```

### 説明

**/etc/nfs.clean** コマンドは、NFS、NIS、またはこの両方の操作をシャットダウンするために使用します。このスクリプトは **shutdown** コマンドで使用されますが、NFS のみまたは NIS (NIS+) のみの操作を停止するときに使用できます。デフォルトでは、すべての NFS および NIS デーモンが停止されます。

**nfs.clean** コマンドはデーモンを正しい順序でシャットダウンするので、**stopsrc -g nfs** の代わりにこのコマンドを使用することが推奨されています。 **stopsrc** コマンドには、グループのデーモンを正しい順序で停止するという概念がありません。このため、**statd** と **lockd** の 2 つのデーモンが実行されている場合、**statd** デーモンが **lockd** デーモンより先に停止されることによって問題が発生する場合があります。

### フラグ

- d** サーバー固有のデーモンのみを停止します。クライアントで実行されている可能性のあるデーモンは停止されません。
- y** サーバー固有の NIS (および NIS+) デーモンのみを停止します。このフラグは、**-d** フラグが使用される場合には指定されることが想定されます。
- t** 指定したシステムのみを停止します。**-t nfs** を指定すると、NFS デーモンのみが停止されます。**-t nis** を指定すると、NIS デーモンのみが停止されます。

### 終了状況

- 0 コマンドは正常に完了しました。
- 1 引数エラー。

### 例

1. すべての NFS および NIS デーモンを停止するには、次のように入力します。

```
/etc/nfs.clean
```

2. NFS のみを停止するには、次のように入力します。

```
/etc/nfs.clean -t nfs
```

3. NFS サービス・デーモンのみを停止するには、次のように入力します。

```
/etc/nfs.clean -d -t nfs
```

### 位置

**/etc/nfs.clean**

### 関連情報

shutdown コマンド。



---

## nfs4cl コマンド

### 目的

現行の NFSv4 統計情報およびプロパティを表示または変更します。

### 構文

```
/usr/sbin/nfs4cl [subcommand] [path] [argument]
```

### 説明

**nfs4cl** コマンドを使用して、クライアントに関するすべての **fsid** 情報を表示するか、**fsid** のファイルシステム・オプションを変更します。

注: **nfs4cl** 更新は、ファイルシステム内の新たにアクセスされるファイルに影響を及ぼします。以前にアクセスされたすべてのファイルに影響を及ぼすには、アンマウントおよび再マウントが必要です。

### サブコマンド

#### resetfsoptions サブコマンド

このサブコマンドは、**fsid** のすべてのオプションをリセットしてデフォルト・オプションに戻します。

注: **cio** および **dio** オプションは **resetfsoptions** サブコマンドを使用してリセットできますが、**cio** および **dio** の動作は、NFS ファイルシステムがアンマウントされてから再マウントされるまで、実際にはオフにされません。

#### setfsoptions サブコマンド

このサブコマンドは、パスおよび引数を取得します。パスはターゲットの **fsid** 構造を指定し、引数はファイルシステム・オプションです。これにより、引数によって指定されたオプションを使用するための、内部 **fsid** が設定されます。指定できる引数のリストを以下に示します。

<b>rw</b>	このパス ( <b>fsid</b> ) にバインドするファイルまたはディレクトリーが、読み取り可能かつ書き込み可能であることを示します。
<b>ro</b>	このパス ( <b>fsid</b> ) にバインドするファイルまたはディレクトリーが、読み取り専用であることを示します。
<b>acdirmax</b>	ディレクトリー属性キャッシュ・タイムアウト値の上限を指定します。
<b>acdirmin</b>	ディレクトリー属性キャッシュ・タイムアウト値の下限を指定します。
<b>acregmax</b>	ファイル属性キャッシュ・タイムアウト値の上限を指定します。
<b>acregmin</b>	ファイル属性キャッシュ・タイムアウト値の下限を指定します。
<b>cio</b>	並行リーダーおよびライターにマウントされるファイルシステムを指定します。このファイルシステム内のファイル上での入出力は、ファイルが <b>open()</b> システム・コール内で指定された <b>O_CIO</b> でオープンされたかのように動作します。
<b>dio</b>	ファイルシステムにおける入出力が、すべてのファイルが <b>open()</b> システム・コールで指定された <b>O_DIRECT</b> でオープンされたかのように動作することを指定します。
<b>hard</b>	この <b>fsid</b> がハード・マウント・セマンティクスを使用することを指定します。
<b>intr</b>	<b>fsid</b> 操作が割り込み可能であることを指定します。
<b>maxpout=value</b>	このファイルシステム上のファイルの、スレッドをスリープにすべきページアウト・レベルを指定します。 <b>maxpout</b> を指定した場合は、 <b>minpout</b> も指定する必要があります。この値は負数であってはならず、 <b>minpout</b> より大でなければなりません。デフォルトは、カーネル <b>maxpout</b> レベルです。

<b>minpout</b> = <i>value</i>	このファイルシステム上のファイルの、スレッドを作動可能にすべきページアウト・レベルを指定します。 <b>minpout</b> を指定した場合は、 <b>maxpout</b> も指定する必要があります。この値は負数であってはなりません。デフォルトは、カーネル <b>minpout</b> レベルです。
<b>noac</b>	属性キャッシュを使用しません。
<b>nocto</b>	クローズとオープン間の整合性を指定しません。
<b>nointr</b>	fsid 中断不可能であることを指定します。
<b>prefer</b> = <i>servername</i>	データが複数のサーバー位置に存在する場合に使用する優先サーバーを、管理上設定します。サーバー名は、ショート・ネーム、ロング・ネーム、IPv4、または IPv6 のいずれのフォーマットでも構いませんが、 <b>nfs4cl</b> コマンドが実行されるときに、クライアントがサーバー名を解決できなければなりません。
<b>rbr</b>	読み取り時に後で解放する (release-behind-when-reading) 機能を使用します。このファイルシステムでファイルの順次読み取りが検出される場合、実メモリー・ページが内部バッファにコピーされると、ファイルに使用されたこのページが解放されます。
<b>rsize</b>	サーバーに対する RPC 呼び出しの読み取りサイズを指定します。
<b>retrans</b>	ソフト・セマンティクスと共に試行する RPC 再送回数を指定します。
<b>soft</b>	ソフト・マウント・セマンティクスを使用する、fsid 操作を指定します。
<b>timeo</b>	サーバーに対する RPC 呼び出しのタイムアウト値を指定します。
<b>wsize</b>	サーバーに対する RPC 呼び出しの書き込みサイズを指定します。
<b>nodircache</b>	ディレクトリー・キャッシュを使用しません。

## showfs サブコマンド

このサブコマンドは、クライアントが現在アクセスしているサーバーに関するファイルシステム固有情報を表示します。情報には、サーバー・アドレス、リモート・パス、「fsid」、およびローカル・パスが含まれます。パスが提供される場合は、fs\_locations オプションおよび fsid オプションなどの追加情報が表示されます。

## showstat サブコマンド

このサブコマンドは、クライアント上にある各 fsid ごとに **df** コマンドが印刷する内容と同じ情報を表示します。この情報には、「Filesystem」、「512-blocks」、「Free」、「%Used」、「Iused」、「%Iused」、および「Mounted on」などのフィールドが含まれます。

## delegreturn サブコマンド

このサブコマンドは、その入力引数としてファイル・パスを受け入れます。このサブコマンドを使用すると、システム管理者は NFS v4 クライアントに指示して、入力パス名で指定したファイルの委任を戻します。

## help サブコマンド

このサブコマンドを使用して、使用方法を印刷します。

## 終了状況

0	コマンドは正常に実行されました。
>0	エラーが発生しました。

## 例

1. クライアント上のすべての fsid 構造を表示するには、次のように入力します。

```
nfs4cl showfs
```

2. **/mnt/usr/sbin** のファイルシステム・オプションを、「retrns=3」のみを含むように設定するには、次のように入力します。

```
nfs4cl setfsoptions /mnt/usr/sbin retrns=3
```

3. **/mnt/use/sbin** のファイルシステム・オプションをリセットするには、次のように入力します。

```
nfs4cl resetfsoptions /mnt/user/sbin
```

4. **/mnt/usr/sbin** の **df** コマンド出力を表示するには、次のように入力します。

```
nfs4cl showstat /mnt/usr/sbin
```

5. **/mnt/usr/sbin** で複製が発生したときにクライアントをサーバー boo にフェイルオーバーさせるには、以下のように入力します。

```
nfs4cl setfsoptions /mnt/usr/sbin prefer=boo
```

## 位置

**/usr/sbin/nfs4cl**

## 関連情報

85 ページの『nfsstat コマンド』, 62 ページの『nfso コマンド』.

---

## nfs4smctl コマンド

### 目的

NFSv4 状態の取り消しを管理します。

### 構文

```
/usr/sbin/nfs4smctl -r [hostname | IP_address]
```

### 説明

NFS v4 状態の取り消しを管理します。

### フラグ

**-r** *hostname*            *hostname* パラメーターまたは *IP\_address* パラメーターのいずれかを使用して状態を取り消す必要のあるクライアントを、指定します。  
*IP\_address*

### ファイル

**/usr/sbin/nfs4smctl**                    **nfs4smctl** コマンドの位置。

### 関連情報

**nfs4cl** コマンド。

---

## nfsd デーモン

### 目的

クライアントからのファイルシステム操作の要求に対してサービスを提供します。

### 構文

```
/usr/sbin/nfsd [ -a | -p { tcp | udp } ] [ -c max_connections ] [ -gp on | off ] [ -gpx count ] [ -gpbypass ] [ -w max_write_size ] [ -r max_read_size ] [ -root directory ] [ -public directory ]  
nservers
```

```
/usr/sbin/nfsd -getnodes
```

```
/usr/sbin/nfsd -getreplicas
```

### 説明

**nfsd** デーモンはサーバー上で実行し、ファイルシステム操作に対するクライアントの要求を処理します。

各デーモンは、一度に 1 つずつ要求を処理します。サーバーが処理すると予測されるロードに基づいてスレッドの最大数を割り当ててください。

**nfsd** デーモンは、次のシステム・リソース・コントローラー(SRC) コマンドを使用して始動および停止されます。

```
startsrc -s nfsd
```

```
stopsrc -s nfsd
```

SRC コマンドで始動するデーモンの数を変更するには、**chnfs** コマンドを使用します。SRC で制御するデーモンのパラメーターを変更する場合は、**chssys** コマンドを使用します。

**注:** **nfsd** デーモンの数が当クライアントに対してサービスするためには不十分である場合、クライアントに対して非幂等操作エラーが戻されます。例えば、クライアントがディレクトリーを除去すると、サーバー上のディレクトリーが除去されても、ENOENT エラーが戻されます。

### フラグ

<b>-a</b>	UDP および TCP トランスポートが処理されることを指定します。
<b>-c max_connections</b>	NFS サーバーで許可される TCP 接続の最大数を指定します。
<b>-gp on off</b>	NFSv4 猶予期間の使用可能化を制御します。有効な値は、on または off です。 <b>-gp</b> オプションが指定されていない場合は、デフォルトにより猶予期間が使用不可にされます。
<b>-gpbypass</b>	NFSv4 猶予期間のバイパスを制御します。このオプションが指定されていると、 <b>-gp</b> オプションがどのように指定されているかに関係なく、猶予期間がバイパスされます。
<b>-gpx count</b>	NFSv4 猶予期間の自動延長を制御します。 <i>count</i> パラメーターは、猶予期間に対する自動延長の総数を指定します。 <b>-gpx</b> オプションを指定しないと、許される自動延長の数はデフォルトで 1 となります。1 つの延長で NFSv4 のリース期間よりも長く猶予期間を延長することはできません。NFSv4 サブシステムは、ランタイム・メトリック (最後に成功した NFSv4 再利用操作の時間など) を使用して、進行中の状態の再利用を検出し、与えられた反復回数 of 期間の範囲を限度に猶予期間を延長します。

<i>nserver</i> s	NFS サーバーが処理できる複数の要求の最大数を指定します。この並行性は、NFS サーバー内のスレッドの動的管理によって、最大限まで達成されます。デフォルトの最大数は 3891 です。 <b>chnfs</b> 、 <b>chssys</b> 、または <b>nfso</b> コマンドを使用して、最大数を変更します。デフォルトの最大設定値の変更は、サーバーのパフォーマンスを制約する可能性があるため、お勧めできません。
<b>-p</b> <i>tcp</i> または <b>-p</b> <i>udp</i>	UDP と TCP をともに、NFS クライアントにトランスポートします (デフォルト)。UDP か TCP しか指定できません。例えば、 <b>-p tcp</b> が使用されると、NFS サーバーは TCP プロトコルを使用して、NFS クライアント要求のみを受信します。
<b>-r</b> <i>max_read_size</i>	NFS バージョン 3 のファイル読み取り要求に対して許可される最大サイズを指定します。デフォルトおよび最大許可値は 32K です。
<b>-w</b> <i>max_write_size</i>	NFS バージョン 3 のファイル書き込み要求に対して許可される最大サイズを指定します。デフォルトおよび最大許可値は 32K です。
<b>-root</b> <i>directory</i>	ルート・ノード NFS バージョン 4 エクスポート・ファイルシステムになるべきディレクトリーを指定します。デフォルトでは、ルート・ノードは / です。ルート・ノードが / 以外のもに設定されている場合は、 <b>chnfs -r</b> を使用して、ノードを / にリセットしてください。このフラグが使用可能なのは、ルート・ノードを変更するために <b>nfsd</b> が実行中だが、現在エクスポートされているファイルシステムがない場合だけです。このフラグは、将来のリリースで除去される可能性があります。代わりに <b>chnfs -r</b> を使用してください。
<b>-public</b> <i>directory</i>	NFS バージョン 4 エクスポート・ファイルシステムのパブリック・ノードになるべきディレクトリーを指定します。デフォルトでは、パブリック・ノードはルート・ノードと同じです。このフラグは、パブリック・ノードを変更するために <b>nfsd</b> が実行中の時に使用できます。パブリック・ノードは、ルート・ノードの子孫でなければなりません。このフラグは、将来のリリースで除去される可能性があります。代わりに <b>chnfs -p</b> を使用してください。
<b>-getnodes</b>	NFS バージョン 4 サーバーの、現行ルート・ノードおよびパブリック・ノードを表示します。このオプションが原因で、NFS サーバー・デーモンが始動することはありません。
<b>-getreplicas</b>	現在の複製使用可能化モードを表示します。レプリカが <b>nfsroot</b> に対して指定されている場合は、そのレプリカが表示されます。

## パラメーター

変更可能なパラメーターは次のとおりです。

*NumberOfNfsds*                      始動させるデーモン数を指定します。このパラメーターは、AIX バージョン 4.2.1 以降には適用されません。

## 例

1. **src** コマンドを使って **nfsd** デーモンを始動するには、次のように入力します。

```
startsrc -s nfsd
```

この例では、startsrc -s nfsd エントリーがスクリプト内で指定された数のデーモンを始動します。

2. ユーザーのシステム上で実行しているデーモンの数を変更するには、次のように入力します。

```
chssys -s nfsd -a 6
```

この例では、chssys コマンドはシステムで実行中の nfsd デーモンの数を 6 にします。

## 関連情報

**chnfs** コマンド、**chssys** コマンド。

**biod** デーモン、**mountd** デーモン。

AIX 5L バージョン 5.3 ナショナル・ランゲージ・サポート ガイドおよびリファレンスのネットワーク・ファイルシステム (NFS) の概要のセクション。

オペレーティング・システムおよびデバイスのマネージのシステム・リソース・コントローラーのセクション。

ネットワークおよびコミュニケーションのマネージ の NFS コマンドのセクション。

---

## nfshostkey コマンド

### 目的

ネットワーク・ファイルシステム (NFS) サーバーのホスト・キーを構成します。

### 構文

```
nfshostkey -l | -L | {-p principal -f file} | { -a -p principal -i address } | { -d -p principal -i address }
```

### 説明

RPCSEC\_GSS RPC セキュリティーを使用している NFS サーバー (または完全クライアント) は、要求を受け入れるために、ホスト・プリンシパルの信任状を獲得する必要があります。この情報を構成するには、**nfshostkey** コマンドを使用します。

すべての完全クライアントおよび NFS サーバーは、1 次ホスト・プリンシパルを持っている必要があります。**nfshostkey** コマンドが設定するホスト・プリンシパルのフォーマットは次のとおりです。

```
nfs/<fully qualified domain name>
```

1 次ホスト・プリンシパルを設定した後、**nfshostkey** コマンドを使用して、他のネットワーク・アドレス用に追加のホスト・プリンシパルを設定できます。サーバーはアドレスのリストを検索して、着信要求が送信されたアドレスを見つけ、該当するプリンシパルを使用します。何も見つからない場合は、1 次プリンシパルが使用されます。2 次ホスト・プリンシパルには、1 次ホスト・プリンシパルのために渡されたものと同じ **keytab** ファイル内にエントリーがなければなりません。それらのエントリーは、完全クライアントでは使用されません。

### フラグ

<b>-a</b>	新しい 2 次ホスト・プリンシパルを追加します。
<b>-d</b>	2 次ホスト・プリンシパルを削除します。
<b>-f file</b>	ホスト・プリンシパルの <b>keytab</b> ファイルへのパスを指定します。
<b>-i address</b>	2 次プリンシパルに対応する IP アドレスを指定します。
<b>-l</b>	1 次ホスト・プリンシパルおよび <b>keytab</b> をリストします。
<b>-L</b>	1 次ホスト・プリンシパル、 <b>keytab</b> 、および 2 次ホスト・プリンシパルをリストします。
<b>-p principal</b>	このホストのプリンシパルを指定します。

## 例

- 1 次ホスト・プリンシパルを設定するには、次のように入力します。

```
nfshostkey -p <principal> -f <keytab file>
```

- 2 次ホスト・プリンシパルを追加するには、次のように入力します。

```
nfshostkey -a -p <principal> -i <ip address>
```

- 3 次ホスト・プリンシパルを削除するには、次のように入力します。

```
nfshostkey -d -p <principal> -i <ip address>
```

## 関連情報

`/etc/nfs/hostkey` ファイル。

---

## nfshostmap コマンド

### 目的

**nfshostmap** クライアントの、ホストからプリンシパルへのマッピングを管理します。

### 構文

```
/usr/sbin/nfshostmap -a hostname alias1 alias2 | -d hostname | -e hostname alias1 alias2 | -l
```

### 説明

別名として定義されるすべてのホストは、サーバーに対する **kerberos** 要求を組み立てる時に **hostname** として定義されたホストに、マップされることとなります。これは、例えば、サーバーが **wizard.sub.austin.ibm.com** および **wizard.austin.ibm.com** というインターフェースを持つ場合に有用です。すなわち、このサーバーの **kerberos** プリンシパルが **wizard.austin.ibm.com** である場合、クライアント上で実行される **nfshostmap -a wizard.austin.ibm.com wizard.sub.austin.ibm.com** はこの問題を処理します。

これにより、**gssd** デーモンが始動時に読み取る `/etc/nfs/princmap` が変更されます。

### フラグ

- a hostname alias1 alias2** 別名からのマッピングを **hostname** に追加します。
- d hostname** **hostname** のすべての別名を削除します。
- e hostname alias1 alias2** **hostname** の以前のマッピングをすべて除去し、指定された別名リストで置き換えます。
- l** システム上のそれぞれのファイルの既存の状態を印刷します。

## 関連情報

`/etc/nfs/princmap` ファイル。



---

## nfso コマンド

### 目的

ネットワーク・ファイルシステム (NFS) のチューニング・パラメーターを管理します。

### 構文

```
nfso [ -p | -r ] [ -c ] { -o Tunable[ =newvalue ] }
```

```
nfso [ -p | -r ] { -d Tunable }
```

```
nfso [ -p | -r ] -D
```

```
nfso [ -p | -r ] -a [ -c ]
```

```
nfso -h [ Tunable ]
```

```
nfso -l [ hostname ]
```

```
nfso -L [ Tunable ]
```

```
nfso -x [ Tunable ]
```

注: 複数のフラグ **-o**、**-d**、**-x**、および **-L** を指定できます。

### 説明

ネットワーク・ファイルシステムのチューニング・パラメーターを構成するには、**nfso** コマンドを使用します。**nfso** コマンドは、ネットワーク・ファイルシステムのチューニング・パラメーターの現行値または次のブート値を設定または表示します。このコマンドは、永久的な変更を行うことも、次のリブートまで変更を遅らせることもできます。このコマンドでパラメーターを設定するか、表示するかは、指定するフラグによって決まります。**-o** フラグを指定すると、設定および表示を行います。つまり、パラメーターの値を表示するかまたはパラメーターに新しい値を設定するかのいずれでも行うことができます。

### チューナブル・パラメーターの変更の影響を理解する

このコマンドの使用には注意してください。**nfso** コマンドを正しく使用しない場合、システムが操作不能になることがあります。

チューナブル・パラメーターを変更する前に、その目的を完全に理解するために、まず下記のチューナブル・パラメーターのセクションで、チューナブル・パラメーターの特性のすべてについて注意して読み、「参照」ポインターがあればこれに従ってください。

その後、このパラメーターの「診断」セクションおよび「チューニング」セクションが本当にご使用の状態に適用されるか、およびこのパラメーターの値の変更がシステムのパフォーマンスを改善する援助となるか、確認する必要があります。

診断セクションとチューニング・セクションの両方に「N/A」とのみ記されている場合は、AIX の開発側から特に指示がないかぎり、多くの場合、このパラメーターを変更する必要はありません。



## フラグ

- a** すべてのチューナブル・パラメーターの現行値、リポート値 (**-r** を指定した場合) または永久値 (**-p** を指定した場合) を、 *Tunable = Value* の対にして、1 行に 1 つずつ表示します。永久オプションの場合、パラメーターの値はそのリポート値と現行値が等しい場合のみ表示されます。そうでない場合は値として **NONE** が表示されます。
- c** **nfso** コマンドの出力フォーマットをコロンで区切るフォーマットに変更します。
- d *Tunable*** *Tunable* 変数をデフォルト値に戻します。 *Tunable* を変更する必要がある (つまり、現在、デフォルト値に設定されていない) 場合、かつタイプが **Bosboot** または **Reboot** であるか、またはタイプが **Incremental** でデフォルト値が変更されており、かつ **-r** が指定されていない場合、変更は行なわれず、警告が表示されます。
- D** すべての *Tunable* 変数をデフォルト値に戻します。変更する必要がある *Tunable* がタイプ **Bosboot** または **Reboot** であるか、またはタイプが **Incremental** でデフォルト値が変更されていて **-r** フラグが指定されていない場合、変更は行なわれず、警告が表示されます。
- h [*Tunable*]** *Tunable* パラメーターのヘルプを表示します (このパラメーターが指定されている場合)。このパラメーターが指定されていない場合は、 **nfso** コマンド使用ステートメントを表示します。
- L [*Tunable*]** 1 つまたはすべての *Tunable* の特性を、次の形式で 1 行に 1 つずつリストします。

NAME	DEPENDENCIES	CUR	DEF	BOOT	MIN	MAX	UNIT	TYPE
portcheck		0	0	0	0	1	On/Off	D
udpchecksum		1	1	1	0	1	On/Off	D
nfs_socketsize		600000	600000	600000	40000	1M	Bytes	D
nfs_tcp_socketsize		600000	600000	600000	40000	1M	Bytes	D

...  
 where:  
 CUR = current value  
 DEF = default value  
 BOOT = reboot value  
 MIN = minimal value  
 MAX = maximum value  
 UNIT = tunable unit of measure  
 TYPE = parameter type: D (for Dynamic), S (for Static), R (for Reboot),  
 B (for Bosboot), M (for Mount), I (for Incremental), C (for Connect), and d (for Deprecated)  
 DEPENDENCIES = list of dependent tunable parameters, one per line

- l *hostname*** システム管理者が **NFS** サーバー上で **NFS** ファイル・ロックを解除できるようにします。 *hostname* 変数は、**NFS** サーバー上でファイル・ロックが保持されている **NFS** クライアントのホスト名を指定します。 **nfso -l** コマンドは、**NFS** サーバーの **rpc.lockd** ネットワーク・ロック・マネージャーへのリモート・プロシージャラー・コールを実行して *hostname* **NFS** クライアントが保持しているファイル・ロックの解除を要求します。

ファイル・ロックを **NFS** サーバー上に保持している **NFS** クライアントがあり、このクライアントがネットワークから切断されているためにリカバリーできない場合、他の **NFS** クライアントが類似するファイル・ロックを取得できるように、 **nfso -l** コマンドを使用して問題のロックを解除することができます。

- o *Tunable* [ =*newvalue* ]** 注: **nfso** コマンドは、ローカル **NFS** サーバーのロック解除に対してのみ使用可能です。値を表示するか、 *Tunable* を *newvalue* に設定します。パラメーターを変更する必要があり (指定された値が現行値と異なる)、このタイプが **Bosboot** または **Reboot** であるか、またはタイプが **Incremental** でその現行値が指定された値より大きく、かつ **-r** が指定されていない場合、変更は行なわれず、警告が表示されます。

新しい値を指定せずに **-r** を指定すると、 *Tunable* の次のブート値が表示されます。 *newvalue* を指定せずに **-p** を指定すると、 *Tunable* の現行値と次のブート値が同じ場合のみ、値が表示されます。そうでない場合は値として **NONE** が表示されます。

**-p** **-o**、**-d** または **-D** を指定した場合、現行値とリポート値の両方を変更します。つまり、現行値の更新だけでなく、**/etc/tunables/nextboot** ファイルの更新をオンにします。Reboot および Bosboot タイプのパラメーターでは現行値を変更できないため、これらの組み合わせは使用できません。

新しい値を指定せずに **-a** または **-o** を使用した場合、パラメーターの現行値と次のブート値が同じ場合のみ、値が表示されます。そうでない場合は値として **NONE** が表示されます。

**-r** **-o**、**-d** または **-D** を指定した場合、リポート値を変更します。つまり、**/etc/tunables/nextboot** ファイルの更新をオンにします。タイプ **Bosboot** のパラメーターを変更すると、**bosboot** を実行するというプロンプトがユーザーに出されます。

新しい値を指定せずに **-a** または **-o** を使用すると、パラメーターの現行値ではなく次のブート値が表示されます。

**-x [Tunable]** 1 つまたはすべての Tunable の特性を、次のスプレッドシート形式で 1 行に 1 つずつリストします。

```
tunable,current,default,reboot,min,max,unit,type,{dtunable }
```

where:

```
current = current value
default = default value
reboot = reboot value
min = minimal value
max = maximum value
unit = tunable unit of measure
type = parameter type: D (for Dynamic), S (for Static), R (for Reboot),
      B (for Bosboot), M (for Mount), I (for Incremental),
      C (for Connect), and d (for Deprecated)
dtunable = space separated list of dependent tunable parameters
```

タイプ **Mount** のパラメーターを変更すると (**-o**、**-d**、または **-D** を使用)、その変更は将来のマウントについてのみに有効であることを示す警告メッセージが表示されます。

タイプ **Connect** のパラメーターを変更すると (**-o**、**-d**、または **-D** フラグを使用)、**inetd** が再始動され、その変更は将来のソケット接続についてのみに有効であることを示す警告メッセージが表示されます。

**-r** を指定せずに、タイプ **Bosboot** または **Reboot** のパラメーターを変更しようとする (**-o**、**-d**、または **-D** を使用)、エラー・メッセージが出されます。

タイプ **Incremental** のパラメーターに、新しい値として現行値よりも小さい値を指定して現行値を変更しようとする (**-o**、**-d**、または **-D** を指定し、**-r** を指定しない)、エラー・メッセージが出されます。

## チューナブル・パラメーターのタイプ

チューニング・コマンド (**no**、**nfso**、**vmo**、**ioo**、**schedo**、および **raso**) で取り扱われるすべてのチューナブル・パラメーターは、下記のカテゴリーに分類されています。

Dynamic	パラメーターをいつでも変更できる場合
Static	パラメーターをいかなる時にでも変更できない場合
Reboot	パラメーターをリポート時にのみ変更できる場合
Bosboot	<b>bosboot</b> を実行してマシンをリポートする場合にのみパラメーターを変更できる場合
Mount	パラメーターの変更が将来のファイルシステムまたはディレクトリーのマウントにのみ有効である場合

Incremental	ブート時を除き、パラメーターを増やすことだけが可能な場合
Connect	パラメーターの変更が将来のソケット接続にのみ有効である場合
Deprecated	このパラメーターの変更が AIX の現行リリースでサポートされなくなっている場合

タイプ **Bosboot** のパラメーターの場合は、変更が加えられるたびに、チューニング・コマンドは自動的に、**bosboot** コマンドを実行したいかどうかを尋ねるプロンプトを出します。タイプ **Connect** のパラメーターの場合は、チューニング・コマンドは自動的に **inetd** デーモンを再始動します。

**nfso** コマンドにより管理されるパラメーターの現行セットには、**Dynamic**、**Mount**、および **Incremental** のタイプがあります。

## 互換モード

5.2 より前の互換モードで実行する場合 (**sys0** の **pre520tune** 属性で制御される。AIX 5.2 互換モードを参照)、パラメーターのリポート値は、タイプ **Bosboot** を除き、実際には意味がありません。このモードでは、値はブート時には適用されないからです。

5.2 より前の互換モードでは、チューニング・パラメーターへのリポート値の設定は、ブート中に呼び出されるスクリプト内でチューニング・コマンドを呼び出すことにより行います。したがって、タイプ **Reboot** のパラメーターは **-r** フラグなしで設定できるので、既存のスクリプトは従来どおり作動します。

このモードは、マシンを AIX 5L バージョン 5.2 に移行すると、自動的にオンになります。完全なインストールの場合、これは **OFF** になり、パラメーターのリポート値は、リポート中に **/etc/tunables/nextboot** ファイルの内容を適用することにより設定されます。このモードの場合のみ、**-r** および **-p** フラグは完全に機能します。新規 5.2 モードの詳細については、*AIX 5L Version 5.3 Performance Tools Guide and Reference* のカーネル・チューニングのセクションを参照してください。

## チューナブル・パラメーター

### client\_delegation

**目的:** NFS バージョン 4 のクライアント代行サポートを使用可能または使用不可にします。

**値:**

デフォルト: 1

範囲: 0 または 1

タイプ: **Dynamic**

**診断:** 該当しない

**チューニング:**

値 1 を指定すると、クライアント代行サポートが使用可能になります。値 0 を指定すると、クライアント代行サポートが使用不可になります。

## lockd\_debug\_level

目的: **rpc.lockd** のデバッグのレベルを設定します。

値:

デフォルト: 0

有効範囲: 0 から 9

タイプ: Dynamic

診断: 該当しない

チューニング:

該当しない

## nfs\_allow\_all\_signals

目的: NFS サーバーが UNIX 95/98 テストの組み合わせ用のブロックされたロックに関するシグナル処理要件に従うことを指定します。

値:

デフォルト: 0

範囲: 0 または 1

タイプ: Dynamic

診断: 該当しない

チューニング:

値が 1 の場合は **nfs\_allow\_all\_signals** がオンになり、値が 0 の場合はオフになります。

## nfs\_auto\_rbr\_trigger

目的: しきい値オフセットをメガバイト単位で指定します。この値を超えると、NFS ファイルの順次読み取りにより、読み取り後にページがメモリーから解放されることとなります。このオプションは、**rbr mount** オプションが効力を持つ場合は無視されます。

値:

- デフォルト: 0 (システムがしきい値を決定することを示します)
- 範囲: -1 (使用不可であることを示します)、0 から最大 NFS ファイル・サイズ (MB 単位)
- タイプ: Dynamic

診断: 大規模な順次読み取り NFS ファイルが原因で、**vmstat** は高ページング率を表示し、**svmon** は大きなクライアント・ページ数を表示します。

チューニング:

この値はメガバイト数に設定する必要があります。このメガバイト数は、NFS ファイルの順次読み取り時にメモリー内でキャッシュ処理されます。キャッシュに入れるファイル・ページによりメモリーが浪費されないようにするために、このしきい値を超える残りのメモリー・ページは、メモリー・ページの読み取り後に解放されます。

**nfs\_device\_specific\_bufs** (AIXバージョン 4.2.1 以降)

**目的:** このオプションは、NFS サーバーがネットワーク装置からのメモリー割り当てを使用できるようにします (ネットワーク装置がこれをサポートする場合)。

**値:**

デフォルト: 1  
範囲: 0 または 1  
タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

NFS サーバーがこのような特別なメモリー割り当てを使用すると、NFS サーバーの全体的なパフォーマンスに良い影響を与えることができます。デフォルトの 1 は、NFS サーバーが特別なネットワーク装置メモリー割り当てを使用できることを意味します。値 0 を使用すると、NFS サーバーは従来のメモリー割り当てを使用して NFS クライアント要求を処理します。これはネットワーク・インターフェースが管理するバッファなので、DMA のセットアップが必要なく、(通常の mbufs よりも) パフォーマンスが向上します。これをサポートするアダプターとして、Micro Channel ATM アダプターと SP2 スイッチ・アダプターの 2 つがあります。

**nfs\_dynamic\_retrans**

**目的:** いつ NFS 要求をサーバーに再送するかを決める時に、NFS クライアントが動的再送アルゴリズムを使用すべきかどうかを指定します。

**値:**

デフォルト: 1  
範囲: 0 または 1  
タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

この機能をオンにした場合、timeo パラメーターは最初の再送でのみ使用されます。このパラメーターを 1 にすると、NFS クライアントはタイムアウトの扱いを過去の NFS サーバーの応答に基づいて調整します。これにより、使用される転送サイズを調整してタイムアウト値を変化させることができます。これはすべて、NFS サーバーの過去の応答時間を累積した履歴に基づいて行なわれます。ほとんどの場合、このパラメーターを調整する必要はありません。場合によっては、単純なタイムアウト操作の方が NFS クライアントにとって望ましいことがあります。このような場合は、ファイルシステムをマウントする前に値を 0 にしてください。

**参照:** 不必要な再送

## nfs\_gather\_threshold

**目的:** 書き込み収集を行う書き込み要求の最小サイズを設定します。

**値:**

デフォルト: 4096

有効範囲: 512 から 8193

タイプ: Dynamic

**診断:** 以下の 2 つの状態のいずれかが観察される場合、**nfs\_gather\_threshold** のチューニングを行うことが適切とされます。

- RPC 要求への応答が遅れていることが観察される。特にクライアントが順次以外の書き込みを排他的に行なっている場合、または書き込まれているファイルについて、ファイル・ロックがクライアントに保持されて書き込まれている場合。
- クライアントが 4096 より小さい書き込みサイズを使用して書き込みをしており、書き込み収集が働いていない。

**チューニング:**

書き込み収集を使用不可にしている場合は、**nfs\_gather\_threshold** を可能な最大の書き込みより大きい値に変更します。NFS バージョン 2 を実行している AIX バージョン 4 の場合、この値は 8192 です。この値を 8193 に変更すると書き込み収集は使用不可になります。これは、上記のシナリオ (1) で説明されている状態で使用してください。書き込みサイズが小さい (1024 など) ために書き込み収集が行われなくなっている場合は、シナリオ (2) の場合のように、書き込み収集パラメーターを小さな書き込みを収集するように変更します (例えば 1024 に設定)。

## nfs\_iopace\_pages (AIX 4.1)

**目的:** 一時点に、VMM を使用してサーバーに書き戻すようにスケジュールされる NFS ファイル・ページの数指定します。この I/O スケジューリング制御は、ファイルのクローズで、システムが *syncd* デーモンを呼び出す時に起こります。

**値:**

デフォルト: 0 (AIX 4.2.1 より前は 32)

範囲: 0 から 65536

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

アプリケーションが NFS マウントのファイルシステムに大きなファイルを書き込むと、そのファイル・データはファイルのクローズ時に NFS サーバーに書き込まれます。場合によっては、サーバーにそのファイルを書き込むためにリソースが取られ、他の NFS ファイル入出力の妨げになる場合があります。このパラメーターは、サーバーに書き込まれる 4 KB ページの数を **nfs\_iopace\_pages** の値に制限します。NFS クライアントはサーバーへの書き込み用に **nfs\_iopace\_pages** をスケジュールし、これらのページがサーバーに書き込まれるのを待ってから次のページのまとまりをスケジュールします。デフォルト値は通常、ほとんどの環境で十分な値です。NFS クライアント・リソースに大量の競合が発生する場合は、値を減らしてください。競合が少ない場合は、値を増やすことができます。この値が 0 の場合は、書き込まれるデフォルトのページ数は、パフォーマンスを最適化するため、ならびに他の NFS ファイル入出力の妨げになる可能性があるリソースの浪費を抑制するために、発見的方法で計算されます。

## nfs\_max\_connections

**目的:** サーバーに可能な TCP 接続の最大数を指定します。

**値:**

デフォルト: 0 (制限なしを意味する)

範囲: 0 10 10000

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

ロードを減らすために、サーバーへの接続の数を制限します。



## nfs\_max\_read\_size

**目的:** 優先的な最大読み取りサイズを設定します。

**値:**

デフォルト: 32768 バイト

有効範囲: 512 から 65536 (TCP による NFS V3 の場合)

512 から 61440 (UDP による NFS V3 の場合)

512 から 8192 (NFS V2 の場合)

タイプ: Dynamic

**診断:** すべてのクライアントの読み取り/書き込みサイズを変更する必要がある場合に役立ちます。これらのクライアントを変更することは非実用的です。デフォルトは、クライアント・マウントで使用される値を使用することを意味します。

**チューニング:**

クライアント内でマウントが直接操作できないとき、特に、読み取り/書き込みサイズが、デフォルトの 32KB であるパケットをドロップするネットワーク上の NIM インストール時には、チューニングにより V3 読み取り/書き込みサイズを小さくしなければならない場合があります。この場合、最大サイズを、ネットワーク上で機能する小さいサイズに設定してください。

ネットワーク・デバイスがパケットをドロップし、サーバーとの通信に包括的な変更が必要な場合も、チューニングが役立ちます。

## nfs\_max\_threads (AIX バージョン 4.2.1 以降)

**目的:** 着信する NFS 要求をサービスするために作成される NFS サーバー・スレッドの最大数を指定します。

**値:**

デフォルト: 3891

範囲: 1 から 3891

タイプ: Dynamic

**診断:** AIX 4.2.1 では、NFS サーバーはマルチスレッドです。NFS サーバー・スレッドは NFS サーバーへの要求が増加するにつれて作成されます。NFS サーバー・スレッドがアイドル状態になると、スレッドは終了します。これにより、サーバーは NFS クライアントのニーズに対応できます。**nfs\_max\_threads** パラメーターは、作成できるスレッドの最大数です。

**チューニング:**

通常、最大値を大きく設定しても、NFS サーバーは必要に応じてスレッドを作成するため、システム全体のパフォーマンスは低下しません。ただし、これはマシンの主な目的が NFS サービスにあると想定した場合です。システムをその他の活動と共用する場合は、スレッドの最大数を低くする必要があるでしょう。最大数は nfsd デーモンのパラメーターとして指定することもできます。

**参照:** 必要な nfsd スレッドの数



## nfs\_max\_write\_size

### 目的:

システム管理者がサーバーの NFS RPC サイズを制御できるようにします。

### 値:

デフォルト: 32768 バイト

有効範囲: 512 から 65536 (TCP による NFS V3 の場合)

512 から 61440 (UDP による NFS V3 の場合)

512 から 8192 (NFS V2 の場合)

タイプ: Dynamic

**診断:** すべてのクライアントの読み取り/書き込みサイズを変更する必要がある場合に役立ちます。これらのクライアントを変更することは非実用的です。デフォルトは、クライアント・マウントで使用される値を使用することを意味します。

### チューニング:

クライアント内でマウントが直接操作できないとき、特に、読み取り/書き込みサイズが、デフォルトの 32KB であるパケットをドロップするネットワーク上の NIM インストール時には、チューニングにより V3 読み取り/書き込みサイズを小さくしなければならない場合があります。この場合、最大サイズを、ネットワーク上で機能する小さいサイズに設定してください。ネットワーク・デバイスがパケットをドロップし、サーバーとの通信に包括的な変更が必要な場合も、チューニングが役立ちます。

## nfs\_repeat\_messages (AIX バージョン 4)

### 目的:

重複する NFS メッセージを検査します。このオプションを使用すると、重複する NFS メッセージは表示されなくなります。

### 値:

デフォルト: 0 (no)

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

### チューニング:

このパラメーターのチューニングはパフォーマンスに影響しません。

**nfs\_v4\_fail\_over\_timeout** (AIX 5.3  
(5300-03 適用) およびそれ以降)

**目的:** データが複製され、現在の関連サーバーがアクセス不能である場合に、他のサーバーに切り替えるまでに NFS クライアントが待つ時間(秒単位)を指定します。デフォルト値 0 が設定されている場合は、クライアントは、マウント時にまたは **nfs4cl** を使用して設定された RPC 呼び出しタイムアウトの 2 倍のタイムアウトを動的に決定します。**nfs\_v4\_fail\_over\_timeout** オプションはクライアント全体にわたるものです。**nfs\_v4\_fail\_over\_timeout** オプションが設定された場合は、このオプションにより、すべての複製データに関するデフォルトの動作が指定変更されます。このオプションは、NFS バージョン 4 にのみ適用されます。

**値:**

デフォルト: 0

範囲: 0 から 4294967295

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

値 0 により、クライアントはタイムアウト値を内的に決定することができます。正の値により、デフォルトを指定変更し、クライアントがアクセスするすべてのデータの複製フェイルオーバー・タイムアウトを秒単位で指定します。

**nfs\_rfc1323** (AIX バージョン 4.3)

**目的:** 非常に大きな TCP ウィンドウ・サイズ (65535 バイトより大きい) のネゴシエーションをシステム間で行なえるようにします。

**値:**

デフォルト: 0

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

NFS クライアントとサーバー間で TCP トランスポートを使用し、両方のシステムがこれをサポートする場合、システムは TCP ウィンドウ・サイズをネゴシエーションして、クライアントとサーバー間でより多くのデータを転送 (*in-flight*) できるようになります。これにより、クライアントとサーバー間で可能なスループットが増えます。**no** コマンドの **rfc1323** オプションと異なり、これは NFS だけに影響し、システム内の他のアプリケーションには影響しません。値 0 はこれが使用不可であることを意味し、値 1 は使用可能であることを意味します。**no** コマンドのパラメーター **rfc1323** が既に設定されている場合、この NFS オプションを設定する必要はありません。

## nfs\_server\_base\_priority

**目的:** `nfsd` デーモンの基本優先順位を設定します。

**値:**

デフォルト: 65

範囲: 31 から 125

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

デフォルトでは、`nfsd` デーモンは浮動する処理優先順位を使用して実行されます。したがって、デーモンの累積 CPU 時間が増えると、デーモンの優先順位は変わります。このパラメーターは `nfsd` デーモンに静的なパラメーターを設定するために使用できます。値 0 は浮動する優先順位 (デフォルト) を表します。許容範囲内のその他の値は、NFS 要求をサーバーで受信した時の `nfsd` デーモンの優先順位の設定に使用されます。このオプションは、NFS サーバーの負荷がシステムにとって過剰になっている場合に使用できます (低くするか、または `nfsd` デーモンを減らすことが望ましい)。また、`nfsd` デーモンをサーバー上で最も優先させる処理の 1 つにしたい場合にも使用できます。このパラメーターを設定すると、その他のプロセスはシステムをほとんど使用できないことになりがちなので、注意してください。この状態は、NFS サーバーが非常に忙しく、サーバー上でその他のプロセスが実行時間を持ってないように実質的に締め出す場合に起こります。

**nfs\_server\_cread** (AIX バージョン 4.2.1 以降)

**目的:** このオプションは、NFS サーバーが、ファイルの読み取りを非常に積極的に実行できるようにします。NFS サーバーは、NFS クライアントからの特定の NFS 読み取り要求にしか応答できません。しかし、NFS サーバーは、ファイル内の現在の読み取り要求の直後に存在するデータを読み取ることができます。これは通常、「先読み」と呼ばれます。NFS サーバーは、デフォルトで「先読み」を実行しません。

**値:**

デフォルト: 1  
範囲: 0 または 1  
タイプ: Dynamic

**診断:** ほとんどの NFS サービス環境では、このパラメーターのデフォルト値 (使用可能) が適切です。ただし、NFS サーバー・メモリーの量がファイル・キャッシングに使用できる場合または NFS による読み取りのアクセス・パターンが主としてランダムである場合 (あるいはその両方の場合) は、このオプションを使用不可にする方が適切であることがあります。

**チューニング:**

**nfs\_server\_cread** オプションを使用可能にすると、NFS サーバーは NFS クライアント用の先読みを非常に積極的に行うようになります。値を 1 にすると、積極的な先読みが行なわれます。値を 0 にすると、通常のシステム・デフォルトの先読み方式が使用されます。通常のシステム先読みは VMM により制御されます。AIX 4.2.1 では、より積極的な上位半分の JFS 先読みが採用されました。このメカニズムは、順序を狂わす要求による先読みの中断 (これは NFS サーバーではよくあることです) の影響を受けにくくなっています。このメカニズムを活動化すると、1 つのクラスター全体 (128 KB の LVM 論理トラック・グループ・サイズ) を読み取ります。

**nfs\_setattr\_error** (AIX バージョン 4.2.1 以降)

**目的:** これを使用可能にすると、NFS サーバーは無効な setattr 要求を無視します。

**値:**

デフォルト: 0 (使用不可)  
範囲: 0 または 1  
タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

このオプションはある種の PC アプリケーション用に提供されています。このパラメーターのチューニングはパフォーマンスに影響しません。

## nfs\_socketsize

**目的:** NFS サーバーの UDP ソケットのキュー・サイズを設定します。

**値:**

デフォルト: 600000

実用的な範囲: 60000 から **sb\_max**

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

UDP のソケット・バッファがいっぱいであるためにパケットがドロップし、nfsd デーモンの数を増やしても役に立たなかったことを netstat が報告した場合に、**nfs\_socketsize** 変数のサイズを増やします。

**参照:** NFS パフォーマンスのモニターおよびチューニングの NFS パフォーマンスに関する TCP/IP チューニング・ガイドライン。

## nfs\_tcp\_duplicate\_cache\_size (AIX バージョン 4.2.1 以降)

**目的:** TCP ネットワーク・トランスポートの NFS サーバーの複製キャッシュに格納するエントリーの数を指定します。

**値:**

デフォルト: 5000

範囲: 1000 から 100000

タイプ: Incremental

**診断:** 該当しない

**チューニング:**

複製キャッシュ・サイズを減らすことはできません。高い処理能力を持つサーバーの複製キャッシュ・サイズを増やします。複製キャッシュは、サーバーが NFS クライアント再送に正しく対応できるようにするために使用されます。クライアントが再送を行う前にサーバーがこのキャッシュをフラッシュすると、サーバーは正しく応答できません。したがって、クライアント再送の前にサーバーが 1000 の操作を処理できる場合は、複製キャッシュ・サイズを増やす必要があります。

NFS サーバーで毎秒受信する NFS 操作の数を計算し、この値を 4 倍します。この結果の値が、NFS サーバーからの正しい応答を可能にするための十分な複製キャッシュ・サイズのはずです。複製キャッシュにより影響される操作は、**setattr()**、**write()**、**create()**、**remove()**、**rename()**、**link()**、**symlink()**、**mkdir()**、**rmdir()** です。

**nfs\_tcp\_socketsize** (AIX バージョン 4.2.1 以降)

**目的:** NFS TCP ソケットのキュー・サイズを設定します。キュー・サイズはバイト数で指定します。TCP ソケットは、送信および受信時に NFS RPC パケットをバッファに入れるために使用します。このオプションは、ソケットの送信および受信ソケット・バッファが使用するメモリーを予約しますが、割り当てはしません。

**値:**

デフォルト: 600000

実用的な範囲: 60000 から **sb\_max**

タイプ: Dynamic

**診断:** 以下の両方の状態が存在する場合、NFS サーバーとクライアント間の順次読み取りまたは書き込みのパフォーマンスは低くなります。

- 大きな (32 KB 以上) RPC サイズが使用されている。
- サーバーとクライアント間の通信が、大きな (9000 バイト以上) MTU サイズを使用するネットワーク・リンクで行われている。

**チューニング:**

**nfs\_tcp\_socketsize** 値は 60 000 より小さい値に設定しないでください。デフォルト値は、大多数の環境に適したものに設定する必要があります。この値は、以下の機能を行うために十分なスペースを確保します。

- TCP ウィンドウ・サイズを制限せずに着信データをバッファに入れる。
- NFS がソケットにデータを書き込める速度を制限せずに発信データをバッファに入れる。

**nfs\_tcp\_socketsize** オプションの値は **sb\_max\_** オプションより小さい値でなければなりません。**sb\_max\_** オプションは **no** コマンドで操作できます。

**参照:** NFS パフォーマンスのモニターおよびチューニング

**nfs\_udp\_duplicate\_cache\_size** (AIX  
バージョン 4.2.1 以降)

**目的:** UDP ネットワーク・トランスポートの NFS サーバーの複製キャッシュに格納するエントリーの数指定します。

**値:**

デフォルト: 5000

範囲: 1000 から 100000

タイプ: Incremental

**診断:** 該当しない

**チューニング:**

複製キャッシュ・サイズを減らすことはできません。高い処理能力を持つサーバーの複製キャッシュ・サイズを増やします。複製キャッシュは、サーバーが NFS クライアント再送に正しく対応できるようにするために使用されます。クライアントが再送を行う前にサーバーがこのキャッシュをフラッシュすると、サーバーは正しく応答できません。したがって、クライアント再送の前にサーバーが 1000 の操作を処理できる場合は、複製キャッシュ・サイズを増やす必要があります。

NFS サーバーで毎秒受信する NFS 操作の数を計算し、この値を 4 倍します。この結果の値が、NFS サーバーからの正しい応答を可能にするための十分な複製キャッシュ・サイズのはずです。複製キャッシュにより影響される操作は、**setattr()**、**write()**、**create()**、**remove()**、**rename()**、**link()**、**symlink()**、**mkdir()**、**rmdir()** です。

**nfs\_use\_reserved\_ports** (AIX バージョン 4.2.1 以降)

**目的:** 予約されていない IP ポート番号を使用することを指定します。

**値:**

デフォルト: 0

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

値 0 は、NFS クライアントが NFS サーバーと通信する時に、予約されていない IP ポート番号を使用します。

**nfs\_v2\_pdots**

**目的:** NFS バージョン 2 マウントのために biod によって使用されるメモリー・プール用のテーブルの数を設定します。

**値:**

デフォルト: 1

範囲: 1 から 8

タイプ: Mount

**診断:** **vmstat -v** を実行して、「**client filesystem I/Os blocked with no fsbuf**」フィールドのゼロ以外の値を探します。

**チューニング:**

ワークロード中にブロック入出力カウントが増分されなくなるまで数値を大きくします。この数値は、**nfs\_v2\_vm\_bufs** と関連して大きくすることが必要な場合があります。

**注:** **bufs** オプションは、**pdts** より前に設定する必要があります。

## nfs\_v2\_vm\_bufs

**目的:** 最初のテーブルの後で作成されたそれぞれの NFS バージョン 2 ページング・デバイス・テーブル (pdt) ごとに使用される、初期空きメモリー・バッファの数を設定します。最初の pdt の設定値は、メモリー・サイズに応じて 1000 または 10000 になります。この初期値は、新しく作成されるそれぞれの pdt のデフォルト値にもなります。

**注:** AIX 5.2 までは、**nfs\_v2\_vm\_bufs** を実行しても、前に設定された pdt に影響を及ぼすことはありませんでした。AIX 5.2 およびそれ以降のリリースでは、**nfs\_v2\_vm\_bufs** を変更すると、現行の NFS バージョン 2 のマウントがないと、古い pdt のサイズにも影響を及ぼします。

### 値:

デフォルト: 1000

範囲: 1000 から 50000

タイプ: Incremental

**診断:** **vmstat -v** を実行して、「**client filesystem I/Os blocked with no fsbuf**」フィールドのゼロ以外の値を探します。

### チューニング:

ワークロード中にブロック入出力カウントが増分されなくなるまで数値を大きくします。この数値は、**nfs\_v2\_pdts** と関連して大きくすることが必要な場合があります。

**注:** **bufs** オプションは、**pdts** より前に設定する必要があります。

## nfs\_v3\_pdts

**目的:** NFS バージョン 3 マウントのために **biod** によって使用されるメモリー・プール用のテーブルの数を設定します。

### 値:

デフォルト: 1

範囲: 1 から 8

タイプ: Mount

**診断:** **vmstat -v** を実行して、「**client filesystem I/Os blocked with no fsbuf**」フィールドのゼロ以外の値を探します。

### チューニング:

ワークロード中にブロック入出力カウントが増分されなくなるまで数値を大きくします。この数値は、**nfs\_v3\_vm\_bufs** と関連して大きくすることが必要な場合があります。

**注:** **bufs** オプションは、**pdts** より前に設定する必要があります。



**nfs\_v3\_server\_readdirplus** (AIX 5.2  
以降)

**目的:** NFS サーバー上の NFS V3 READDIRPLUS 操作を使用可能または使用不可にします。

**値:**

デフォルト: 1 (使用可能)

範囲: 0 から 1

タイプ: Dynamic

**診断:** READDIRPLUS 操作では、NFS V3 マウントを使用する NFS マウント済みのファイルシステムで非常に大きなディレクトリーを読み取るときにオーバーヘッドを追加します。これにより、**nfsd** スレッドによる CPU 消費が過度になり、NFS クライアントによる **ls** などのコマンドへの応答時間が遅くなる可能性があります。

**チューニング:**

READDIRPLUS 操作を使用不可にすると、NFS V3 による非常に大きなディレクトリーを読み取る際のオーバーヘッドの削減に役立ちます。ただし、これは NFS バージョン 3 標準には準拠しないことに注意してください。ほとんどの NFS V3 クライアントは、自動的に READDIR 操作を使用するようにフォールバックしますが、問題が起るようならば、このオプションをデフォルト値に戻すべきです。

**nfs\_v3\_vm\_bufs**

**目的:** 最初のテーブルの後で作成されたそれぞれの NFS バージョン 3 ページング・デバイス・テーブル (pdt) ごとに使用される、初期空きメモリー・バッファの数を設定します。最初の pdt の設定値は、メモリー・サイズに応じて 1000 または 10000 になります。この初期値は、新しく作成されるそれぞれの pdt のデフォルト値にもなりません。

**注:** AIX 5.2 までは、**nfs\_v3\_vm\_bufs** を実行しても、前に設定された pdt に影響を及ぼすことはありませんでした。AIX 5.2 およびそれ以降のリリースでは、**nfs\_v3\_vm\_bufs** を変更すると、現行の NFS バージョン 3 のマウントがないと、古い pdt のサイズにも影響を及ぼします。

**値:**

デフォルト: 1000

範囲: 1000 から 50000

タイプ: Incremental

**診断:** **vmstat -v** を実行して、「**client filesystem I/Os blocked with no fsbuf**」フィールドのゼロ以外の値を探します。

**チューニング:**

ワークロード中にブロック入出力カウントが増分されなくなるまで数値を大きくします。この数値は、**nfs\_v3\_pdots** と関連して大きくすることが必要な場合があります。

**注:** **bufs** オプションは、**pdts** より前に設定する必要があります。

## nfs\_v4\_pdts

**目的:** NFS バージョン 4 マウントのために `biod` によって使用されるメモリー・プール用のテーブルの数を設定します。

**値:**

デフォルト: 1  
範囲: 1 から 8  
タイプ: Mount

**診断:** `vmstat -v` を実行して、「**client filesystem I/Os blocked with no fsbuf**」フィールドのゼロ以外の値を探します。

**チューニング:**

ワークロード中にブロック入出力カウントが増分されなくなるまで数値を大きくします。この数値は、`nfs_v4_vm_bufs` と関連して大きくすることが必要な場合があります。

**注:** `bufs` オプションは、`pdts` より前に設定する必要があります。

## nfs\_v4\_vm\_bufs

**目的:** 最初のテーブルの後で作成されたそれぞれの NFS バージョン 4 ページング・デバイス・テーブル (pdt) ごとに使用される、初期空きメモリー・バッファの数を設定します。最初の pdt の設定値は、メモリー・サイズに応じて 1000 または 10000 になります。この初期値は、新しく作成されるそれぞれの pdt のデフォルト値にもなります。

**注:** AIX 5.2 までは、`nfs_v4_vm_bufs` を実行しても、前に設定された pdt に影響を及ぼすことはありませんでした。AIX 5.2 およびそれ以降のリリースでは、`nfs_v4_vm_bufs` を変更すると、現行の NFS バージョン 4 のマウントがないと、古い pdt のサイズにも影響を及ぼします。

**値:**

デフォルト: 1000  
範囲: 1000 から 50000  
タイプ: Incremental

**診断:** `vmstat -v` を実行して、「**client filesystem I/Os blocked with no fsbuf**」フィールドのゼロ以外の値を探します。

**チューニング:**

ワークロード中にブロック入出力カウントが増分されなくなるまで数値を大きくします。この数値は、`nfs_v4_pdts` と関連して大きくすることが必要な場合があります。

**注:** `bufs` オプションは、`pdts` より前に設定する必要があります。

## portcheck

**目的:** NFS 要求が特権のあるポートからのものかどうかを検査します。

**値:**

デフォルト: 0

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

値 0 は、NFS サーバーが行うポート検査を使用不可にします。値 1 は、着信する NFS 要求のポート検査を NFS サーバーが行うようにします。これは、わずかなパフォーマンス向上を求める構成上の決定です。

## server\_delegation

**目的:** NFS バージョン 4 のサーバー代行サポートを使用可能または使用不可にします。

**値:**

デフォルト: 1

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

値 1 を指定すると、代行サポートが使用可能になります。値 0 を指定すると、代行サポートが使用不可になります。サーバー代行は、**/etc/exports** ファイルおよび **exportfs** を使用して制御することもできます。

## statd\_debug\_level

**目的:** **rpc.statd** のデバッグのレベルを設定します。

**値:**

デフォルト: 0

有効範囲: 0 から 9

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## statd\_max\_threads

**目的:** **rpc.statd** で使用されるスレッドの最大数を設定します。

**値:**

デフォルト: 50

有効範囲: 1 から 1000

タイプ: Dynamic

**診断:** **rpc.statd** は、リモート・マシンとの接続を同時に再確立できるように、マルチスレッド化されます。**rpc.statd** スレッドは、要求が増すたびに作成されます。通常、これは **rpc.statd** が通信できないマシンとの接続を再確立しようとするためです。**rpc.statd** スレッドがアイドル状態になると、スレッドは終了します。**statd\_max\_threads** パラメーターは、作成できるスレッドの最大数です。

**チューニング:**

該当しない

## udpchecksum

**目的:** NFS UDP パケットのチェックサムの生成をオンまたはオフにします。

**値:**

デフォルト: 1

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

パケット破壊が起こりうるネットワークにはすべて、この値を必ずオンにしてください。これをオフにすればわずかなパフォーマンス向上がありますが、データ破壊の機会が増えるという損失があります。

## utf8 (AIX 5.3 以降)

**目的:** このオプションは、NFS V4 が UTF8 検査を実行できるようにします。

**値:**

デフォルト: 1

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

値 1 を指定すると、ファイル名の UTF-8 検査がオンになります。  
値 0 を指定すると、UTF-8 検査がオフになります。

## utf8\_validation

**目的:** NFS バージョン 4 クライアントおよびサーバーのファイル名の検査を使用可能にして、ファイル名が UTF-8 仕様に準拠しているか確認します。

**値:**

デフォルト: 1  
範囲: 0 または 1  
タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

値 1 は、ファイル名の UTF-8 検査をオンにします。値 0 はこれをオフにします。

## 例

1. **portcheck** チューナブル・パラメーターの値をゼロにするには、次のように入力します。  
`nfso -o portcheck=0`
2. **udpchecksum** チューナブル・パラメーターの値を次のリポートでデフォルト値の 1 にするには、次のように入力します。  
`nfso -r -d udpchecksum`
3. すべてのチューナブル・パラメーターとその現在値のリストを、コロンの区切ったフォーマットで印刷するには、次のように入力します。  
`nfso -a -c`
4. **nfso** コマンドが管理するすべてのチューナブル・パラメーターの、現行値とリポート値、範囲、単位、タイプと依存関係をリストするには、次のように入力します。  
`nfso -L`
5. **nfs\_tcp\_duplicate\_cache\_size** のヘルプ情報を表示するには、次のように入力します。  
`nfso -h nfs_tcp_duplicate_cache_size`
6. **nfs\_dynamic\_retrans** を永久にオフにするには、次のように入力します。  
`nfso -p -o nfs_dynamic_retrans=0`
7. すべてのネットワーク・ファイルシステム・チューニング・パラメーターのリポート値をリストするには、次のように入力します。  
`nfso -r -a`
8. **nfso** コマンドが管理するすべてのチューナブル・パラメーターの、現行値とリポート値、範囲、単位、タイプと依存関係をリスト (スプレッドシート形式) するには、次のように入力します。  
`nfso -x`

## 関連情報

**netstat** コマンド、**no** コマンド、**vmo** コマンド、**ioo** コマンド、**raso** コマンド、**schedo** コマンド、**tunchange** コマンド、**tunsave** コマンド、**tunrestore** コマンド、**tuncheck** コマンド、**tundefault** コマンド。

ネットワーク・ファイルシステムのセクション。

TCP/IP のセクション。

NFS 統計情報およびチューニング・パラメーターのセクション。

NFS コマンドのセクション。

*AIX 5L Version 5.3 Performance Tools Guide and Reference*のカーネル・チューニングのセクション。

AIX 5.2 互換モードのセクション。

---

## nfsrgyd デーモン

### 目的

NFS V4 または RPCSEC-GSS を使用しているサーバーおよびクライアントからの、名前と ID との変換要求をサービスします。

### 構文

```
nfsrgyd [ -f ] [ -T heartBeatInt ]
```

### 説明

**nfsrgyd** デーモンは、NFS サーバーおよび NFS クライアントに対して、名前変換サービスを提供します。NFS スtring属性と UNIX 数値 ID との間の変換を実行するためには、このデーモンが稼働している必要があります。

環境変数 **NFS\_NOBODY\_USER** および **NFS\_NOBODY\_GROUP** は、名前変換で使用される匿名ユーザーおよびグループ所有者Stringに影響します。これらの環境変数が設定されていない場合、デフォルト値である **nobody** が使用されます。これらは、ファイル **/etc/environment**、または **nfsrgyd** が開始される前にコマンド行で、設定することができます。

**nfsrgyd** デーモンの実行前に、ローカル NFS ドメインが設定されていることが必要です。これは、**chntfsdom** コマンドを使用して設定することができます。

注: **nfsrgyd** デーモンは一時ポートを使用します。

### フラグ

<b>-f</b>	新規プロセスを作成し、名前変換キャッシュおよび出口をフラッシュします。
<b>-T</b>	以降の LDAP サーバー再接続の時間間隔を指定します。有効な値は 60 から 3600 秒です。デフォルト値は 300 です。

### 例

1. **nfsrgyd** デーモンは **/etc/rc.nfs** ファイルから開始されます。以下のシステム・リソース・コントローラー (SRC) コマンドにより、**nfsrgyd** デーモンの開始と停止を行うことができます。

```
startsrc -s nfsrgyd
stopsrc -s nfsrgyd
```

2. **chssys** コマンドを使用して **nfsrgyd** デーモンに渡すパラメーターを変更するには、次のように入力します。

```
chssys -s nfsrgyd -a "-T 360"
```

ヒント: 変更はデーモンを再始動するまで有効になりません。 **nfsrgyd** デーモンが再始動した後、*heartBeatInt* 間隔の値は持続します。

## セキュリティ

ユーザーは **root** 権限が必要です。

## ファイル

**/etc/environment** NFS 環境変数が入っています。

## 関連情報

**chnfsdom** コマンド、**chnfsrtd** コマンド、および **chnfssec** コマンド。

---

## nfsstat コマンド

### 目的

ネットワーク・ファイルシステム (NFS) およびリモート・プロシーチャー・コール (RPC) の呼び出しに関する統計情報を表示します。

### 構文

```
/usr/sbin/nfsstat [ -c ] [ -d ] [ -s ] [ -n ] [ -r ] [ -m ] [ -4 ] [ -z ] [ -t ] [ -b ] [ -g ]
```

### 説明

**nfsstat** コマンドは、カーネルに対する NFS およびリモート・プロシーチャー・コール (RPC) インターフェースに関する統計情報を表示します。また、このコマンドを使用してこの情報を再初期化することができます。フラグを指定しない場合、デフォルトは **nfsstat -csnr** コマンドです。このオプションを指定した場合、コマンドはすべての情報を表示しますが、再初期化は行いません。

### RPC サーバー情報

サーバー RPC ディスプレイには次のフィールドが含まれます。

<b>calls</b>	受信した RPC 呼び出しの合計回数。 <b>-4</b> フラグを使用する場合は、この回数には NFS バージョン 4 呼び出しが含まれます。それ以外の場合は、バージョン 2 とバージョン 3 の合計のみが表示されます。
<b>badcalls</b>	RPC レイヤーがリジェクトした呼び出しの合計回数。 <b>-4</b> フラグを使用する場合は、この回数には NFS バージョン 4 呼び出しが含まれます。それ以外の場合は、バージョン 2 とバージョン 3 の合計のみが表示されます。
<b>nullrecv</b>	受信しようとしたときに RPC 呼び出しが使用可能でなかった回数。
<b>badlen</b>	最小サイズの RPC 呼び出しより短い RPC 呼び出しの回数。
<b>xdrCALL</b>	XDR によるデコードができなかったヘッダーを持つ RPC 呼び出しの回数。
<b>dupchecks</b>	重複する要求キャッシュで検索した RPC 呼び出しの回数。
<b>dupreqs</b>	検出された重複する RPC 呼び出しの回数。

### RPC クライアント情報

**calls** 送信された RPC 呼び出しの合計数。

badcalls	RPC レイヤーがリジェクトした呼び出しの合計数。
badxid	サーバーからの返信が未解決の呼び出しと一致しなかった回数。
timeouts	サーバーからの返信を待つ間に呼び出しがタイムアウトになった回数。
newcreds	認証情報を最新化しなければならなかった回数。
badverfs	応答における正しくない検査子のために呼び出しが失敗した回数。
timers	計算されたタイムアウト値が、呼び出しに対して指定されている最小タイムアウト値と同じかそれより大きかった回数。
cantconn	サーバーへの接続が失敗したために呼び出しが失敗した回数。
nomem	メモリーの割り当てが失敗したために呼び出しが失敗した回数。
interrupts	呼び出しが完了前にシグナルによって割り込まれた回数。
retrans	サーバーからの返信を待つ間にタイムアウトが起こったため、呼び出しを再送しなければならなかった回数。これは、RPC コネクションレス・トランスポートにしか適用できません。
dupchecks	重複する要求キャッシュでルックアップした RPC 呼び出しの数。
dupreqs	検出された重複する RPC 呼び出しの数。

## NFS サーバー情報

NFS サーバーは、受信した NFS 呼び出し数 (calls) とリジェクトされた NFS 呼び出し数 (badcalls)、および実行された各種呼び出しの数とパーセンテージを表示します。

## NFS クライアント情報

NFS クライアントは、送信された呼び出し数とリジェクトされた呼び出し数、CLIENT ハンドルが受信された回数 (clgets)、クライアント・ハンドルに未使用のエントリーがなかった回数 (clatoomany)、および各種呼び出しの数とそれぞれのパーセンテージを表示します。

## NFS レジストリー・デーモン情報

NFS レジストリー・デーモン表示は、クライアントおよびサーバーからの、UID/GID 名とストリング名を変換する要求の回数を表示します。

## -m 情報

**-m** フラグは、**mount** オプションで設定する **mount** フラグの情報、システム内部の **mount** フラグの情報、およびその他の **mount** 情報を表示します。詳細は、**mount** コマンドを参照してください。

次の **mount** オプションは、**mount** フラグで設定します。

auth	次の値のいずれかを指定します。
	<b>none</b> 認証なし。
	<b>unix</b> UNIX スタイル認証 (UID、GID)。
	<b>des</b> des スタイル認証 (暗号化されたタイム・スタンプ)。
hard	ハード・マウント。
soft	ソフト・マウント。
intr	ハード・マウントで割り込みが許可されています。
nointr	ハード・マウントで割り込みが許可されていません。
noac	クライアントが属性をキャッシュしていません。
rsize	バッファー・サイズをバイト単位で読み取ります。
wsize	バッファー・サイズをバイト単位で書き込みます。
retrans	NFS 再送。
nocto	クローズとオープン間の整合性がありません。
llock	ローカル・ロックが使用中です (ロック・マネージャーがありません)。
grpuid	グループ ID 継承。
vers	NFS バージョン。



proto            プロトコル。

次の **mount** オプションは、システム内部のものです。

printed	応答メッセージを出力しません。
down	サーバーをダウンします。
dynamic	動的転送サイズの調整。
link	サーバーがリンクをサポートします。
symlink	サーバーがシンボリック・リンクをサポートします。
readdir	<b>readdirplus</b> のかわりに <b>readdir</b> を使用します。

## -t 情報

**-t** フラグは、NFS ID マッピング・サブシステムの変換要求に関連する情報を表示します。

ids_to_strings	ID からストリングへの変換要求の回数。
strings_to_ids	ストリングから ID への変換要求の回数。
resolve_errors	欠落データが原因で失敗した、変換要求の回数。
badowners	無効な入力の原因で失敗した、変換要求の回数。
cache_hits	変換キャッシュによって処理された変換要求の回数。
cache_misses	変換キャッシュによって処理されなかった変換要求の回数。
cache_entries	変換キャッシュ内のエントリーの数。
cache_recycles	期限切れ変換キャッシュ内のエントリーの数。

## フラグ

- b** NFS バージョン 4 サーバーの追加統計情報を表示します。
- c** クライアント情報を表示します。クライアント・サイドの NFS および RPC 情報だけを表示します。ユーザーは、レポートをクライアント・データだけに制限できます。 **nfsstat** コマンドは、クライアントが送信およびリジェクトした RPC 呼び出しおよび NFS 呼び出しの数に関する情報を提供します。クライアントの NFS 情報または RPC 情報だけを表示するには、このフラグを **-n** オプションまたは **-r** オプションと組み合わせて指定します。
- d** NFS バージョン 4 の代行に関する情報を表示します。

- g** RPCSEC\_GSS 情報を表示します。RPCSEC\_GSS 情報セクションには次が含まれます。
- activegss**  
アクティブな RPCSEC\_GSS コンテキスト
- discardgss**  
廃棄された RPCSEC\_GSS メッセージ
- krb5est**  
確立された krb5 コンテキスト
- krb5iest**  
確立された krb5i コンテキスト
- krb5pest**  
確立された **krb5p** コンテキスト
- expgss**  
期限切れ RPCSEC\_GSS コンテキスト
- badaccept**  
gss\_accept\_sec\_context 失敗
- badverify**  
gss\_verify\_mic 失敗
- badgetmic**  
gss\_get\_mic 失敗
- badwrap**  
gss\_wrap 失敗
- badunwrap**  
gss\_unwrap 失敗
- m** サーバーの名前およびアドレスとともにマウントされたそれぞれの NFS ファイルシステムの統計情報、現在の読み取りサイズおよび書き込みサイズ、再送数、および動的再送に使用したタイマーを表示します。このフラグは、AIX バージョン 4.2.1 以降のバージョンにのみ適用されます。
- n** NFS 情報を表示します。クライアントとサーバーの両方の NFS 情報を出力します。NFS クライアント情報または NFS サーバー情報のいずれかだけを出力するには、このフラグを **-c** オプションや **-s** オプションと組み合わせて指定します。
- r** RPC 情報を表示します。
- s** サーバー情報を表示します。
- t** NFS ID マッピング・サブシステムの変換要求に関連する統計情報を表示します。NFS クライアント情報または NFS サーバー情報のいずれかだけを出力するには、**-c** オプションと **-s** オプションを組み合わせて指定します。
- 4** **-c**、**-n**、**-s**、または **-z** の各フラグと組み合わせると、既存の NFS バージョン 2 とバージョン 3 のデータに追加して、NFS バージョン 4 のクライアントまたはサーバーの情報が含まれます。このオプションが無指定の場合、出力は AIX バージョン 5.3 以前のバージョンにおける **nfsstat** コマンドからの出力と同じになります。
- z** 統計情報を再初期化します。このフラグは root ユーザー専用であり、上記のフラグとともに指定し、特定の統計情報を出力後、ゼロにします。

## 例

1. クライアントが送信およびリジェクトした RPC と NFS の呼び出し回数に関する情報を表示するには、次のように入力します。

```
nfsstat -c
```

2. クライアントの NFS 呼び出し関連情報を表示して出力するには、次のように入力します。

```
nfsstat -cn
```

3. AIX 4.2.1 以降における NFS マウント・ファイルシステムのそれぞれの統計情報を表示します。

```
nfsstat -m
```

4. クライアントとサーバーの RPC 呼び出しに関連する情報を表示して出力するには、次のように入力します。

```
nfsstat -r
```

5. サーバーが受信およびリジェクトした RPC と NFS の呼び出し回数に関する情報を表示するには、次のように入力します。

```
nfsstat -s
```

6. すべての呼び出しに関する情報をクライアントとサーバー上でリセットしてゼロにするには、次のように入力します。

```
nfsstat -z
```

注: **-z** フラグを使用するには、root ユーザー権限が必要です。

## 関連情報

ネットワークおよびコミュニケーションのマネージ のネットワーク・ファイルシステム (NFS) の概要のセクション。

ネットワークおよびコミュニケーションのマネージ の NFS コマンドのリストのセクション。

パフォーマンス・マネージメントの NFS パフォーマンスのセクション。

---

## nice コマンド

### 目的

コマンドを低い優先順位あるいは高い優先順位で実行します。

### 構文

```
nice [ -Increment | -n Increment ] Command [ Argument ... ]
```

### 説明

**nice** コマンドを使用すると、コマンドを通常よりも低い優先順位で実行できます。 *Command* パラメーターは、システム上の実行可能ファイルの名前です。 *Increment* 値を指定しなければ、**nice** コマンドはデフォルトで 10 ずつ増分します。現在よりも高い優先順位でコマンドを実行するには、root ユーザー権限を持っていない限りはなりません。通常、プロセスの優先順位はその **nice** の値で表されます。

**nice** の値として、-20 から 19 までの値を指定することができ、この場合 19 が最も低い優先順位です。例えば、コマンドが通常は優先順位 10 で実行される場合に、増分値 5 を指定すると、そのコマンドは現在よりも低い優先順位 15 で実行され、実行速度が低下します。該当する権限を持たないユーザーがコマンドの優先順位を上げようとしても、**nice** コマンドはエラー・メッセージを戻しません。コマンドの優先順位は変更されず、システムは正常にコマンドを始動します。

**nice** の値は、実行中のプロセスの現在の優先順位を計算するためにシステムに使用されます。 **-l** フラグを指定して **ps** コマンドを使用すると、コマンドの **nice** の値が表示されます。 **nice** の値は、**ps** コマンド出力の **NI** 見出しの下に表示されます。

注: **csch** コマンドに、**nice** という名前の組み込みコマンドが入っています。 **/usr/bin/nice** コマンドと、**csch** コマンドの **nice** コマンドが同じように機能するとは限りません。 **csch** コマンドの **nice** コマンドについては、**csch** コマンドを参照してください。

## フラグ

- Increment** コマンドの優先順位を上方または下方に増分します。正または負の数値を指定できます。正の増分値を指定すると、優先順位が低くなります。負の増分値を指定すると、優先順位が高くなります。負の増分を指定できるのは、**root** 権限を持つユーザーだけです。 **nice** の値が **-20** から **19** までの範囲を超えるような増分値を指定すると、**nice** の値は上限の値に設定されます。このフラグは **-n Increment** フラグと同じです。
- n Increment** このフラグは **-Increment** フラグと同じです。

## 終了状況

**Command** パラメーターで指定されたコマンドを開始すると、**nice** コマンドの終了状況は、**Command** パラメーターで指定されたコマンドの終了状況になります。それ以外の場合、**nice** コマンドは以下のいずれかの値を戻して終了します。

- 1-125** **nice** コマンドでエラーが発生しました。  
**126** **Command** パラメーターで指定されたコマンドを検出しましたが、呼び出すことができませんでした。  
**127** **Command** パラメーターで指定されたコマンドを検出できませんでした。

## 例

- 非常に低い優先順位を指定するには、次のように入力します。

```
nice -n 15 cc -c *.c &
```

この例では、**cc** コマンドが **nice** コマンドで設定したデフォルト優先順位よりも低い優先順位で、バックグラウンドで実行されます。

- 非常に高い優先順位を指定するには、次のように入力します。

```
nice --10 wall <<end  
System shutdown in 2 minutes!  
end
```

この例では、**wall** コマンドがどのユーザー・プロセスよりも高い優先順位で実行され、システムで実行中のその他の作業すべての処理速度が低下します。例の **<<end** および **end** の部分は *here* ドキュメントを定義します。これは最終行の前までに入力したテキストをコマンドの標準入力として使用します。

注: **root** ユーザー権限がないと、このコマンドを実行しても、**wall** コマンドは通常優先順位で実行されます。

- コマンドを低い優先順位で実行するには、次のように入力します。

```
nice cc -c *.c
```

この例で、**cc** コマンドは低い優先順位で実行されます。

注: これは、コマンドをバックグラウンドで実行しません。ワークステーションで他の作業を行うことはできません。

- 優先順位の低いコマンドをバックグラウンドで実行するには、次のように入力します。

```
nice cc -c *.c &
```

この例で、**cc** コマンドは低い優先順位でバックグラウンドで実行されます。**cc** コマンドの実行中に、ワークステーションで他のコマンドを実行できます。バックグラウンド (非同期) 処理については、オペレーティング・システムおよびデバイスのマネージのシェルを参照してください。

## ファイル

`/usr/bin/nice` **nice** コマンドが入っています。

## 関連情報

**csch** コマンド、**nohup** コマンド、**renice** コマンド。

**nice** サブルーチン。

オペレーティング・システムおよびデバイスのマネージのプロセスのセクションで、プロセスおよび入出力リダイレクトについて説明しています。

オペレーティング・システムおよびデバイスのマネージのシェルのセクションで、シェルとは何か、さまざまなタイプ、およびコマンドの解釈の仕方にシェルがどのように影響するかについて説明しています。

パフォーマンス・マネージメントの マイクロプロセッサの競合の制御のセクション。

---

## nim コマンド

### 目的

ネットワーク・インストール管理 (NIM) オブジェクトの操作を行います。

### 構文

```
nim { -o Operation } [ -F ] [ -t Type ] [ -a Attribute=Value . . . ] {ObjectName}
```

### 説明

**nim** コマンドは、NIM オブジェクトで操作を行います。実行する操作のタイプは、*ObjectName* パラメーターで指定するオブジェクト型によって異なります。実行できる操作には、環境の初期化やリソースの管理があります。サポートされている操作のリストを表示するには、**lsnim** コマンドを使用します。

### フラグ

**-aAttribute = Value . . .**

指定された属性に指定された値を割り当てます。**lsnim -q Operation -t Type** コマンドを使用して、特定の操作に有効な属性のリストを入手します。

**-F**

いくつかの安全検査をオーバーライドします。

NIM オブジェクトで行う操作を指定します。実行できる操作は次のとおりです。

**allocate** リソースを割り当てて使用できるようにします。

**alt\_disk\_install**

代替ディスク・インストールを実行します。

**alt\_disk\_mig**

**rootvg**gua のコピーを空きディスク (複数可) に作成し、同時にそれを AIX の新規バージョンまたはリリース・レベルにマイグレーションする。

**bos\_inst** BOS インストールを実行します。

**change** オブジェクトの属性を変更します。

**check** NIM オブジェクトの状況を検査します。

**cust** ソフトウェアのカスタマイズを実行します。

**dbts** シン・サーバーをデバッグします。

**deallocate**

リソースの割り当てを解除します。

**define** オブジェクトを定義します。

**diag** マシンが診断イメージをブートできるようにします。

**dkls\_init** ディスクレス・マシンの環境を初期化します。

**dtls\_init** データレス・マシンの環境を初期化します。

**fix\_query**

指定された APAR またはキーワードの固定情報をリストします。

**linux\_inst**

スタンドアロン・クライアント上に Linux® をインストールします。

**lppchk** インストールされたファイルセットを NIM マシンおよび SPOT で検査します。

**lppmgr** **lpp\_source** 内の不要なソフトウェア・イメージを除去します。

**lspp** オブジェクトのライセンス・プログラム情報をリストします。

**maint** ソフトウェアの保守を行います。

**maint\_boot**

マシンが保守モードでブートできるようにします。

**reboot** NIM クライアント・マシンをリブートします。

**remove** オブジェクトを除去します。

**reset** オブジェクトの NIM 状態をリセットします。

**restvg** **restvg** 操作を実行します。

**select** グループに関して行われている操作から、グループ・メンバーを包含または除外します。

**showlog** NIM クライアントのインストール、ブートまたはカスタマイズのログ、あるいは SPOT のインストール・ログを NIM マスターから表示します。

**showres** NIM リソースの内容を表示します。

**swts** 異なる COSI にシン・サーバーをスイッチします。

**sync** NIM データベースを代替マスターと同期化します。

**sync\_roots**

特定の共用プロダクト・オブジェクト・ツリー (SPOT) 用にディスクレス・クライアントとデータレス・クライアントのルート・ディレクトリーを同期化します。

**unconfig** NIM マスター・ファイルセットを構成解除します。

**update** ソフトウェアを **lpp\_source** に追加するか、またはソフトウェアを **lpp\_source** から除去します。

**lsnim -POt Type** コマンドを使用すると、特定のタイプに有効な操作のリストを入手できます。

定義操作のために NIM オブジェクト型を指定します。指定できるタイプは次のとおりです。

#### リソース・タイプ

##### **adapter\_def**

2 次アダプター定義ファイルを含むディレクトリー。

##### **boot**

ブート・イメージがクライアントに割り振り済みであることを示すのに使用される、内部的に管理された NIM リソース。

##### **bosinst\_data**

基本システムのインストール中に使用する Config ファイル。

##### **dump**

クライアント・ダンプ・ファイルの親ディレクトリー。

##### **exclude\_files**

**mksysb** イメージから除外されるファイルを含みます。このリソースは、AIX バージョン 4.2 以降にだけ適用されます。

##### **fb\_script**

マシンの最初のリポート時に実行される実行可能スクリプト。

##### **fix\_bundle**

**cust** 操作または **fix\_query** 操作の固定 (キーワード) 入力ファイル。

##### **home**

クライアント **/home** ディレクトリーの親ディレクトリー。

##### **image\_data**

基本システムのインストール中に使用する Config ファイル。

##### **installp\_bundle**

**Installp** バンドル・ファイル。

##### **linux\_source**

Linux インストール・メディアを表します。

##### **log**

ネットワーク・インストール中にログ・データを収集します。

##### **lpp\_source**

オプションのプロダクト・イメージのソース・デバイス。

##### **mksysb**

**mksysb** イメージ。

##### **nim\_script**

NIM が NIM 操作の一部としてスクリプトを実行する必要があることを示すのに使用される、内部的に管理された NIM リソース

##### **paging**

クライアント・ページング・ファイルの親ディレクトリー。

##### **root**

クライアント / (ルート) ディレクトリーの親ディレクトリー。

##### **resolv\_conf**

ネーム・サーバー構成ファイル。

##### **savevg**

**savevg** イメージ。

##### **script**

クライアントで実行する実行可能ファイル。

##### **shared\_home**

クライアント間で共用する **/home** ディレクトリー。

##### **spot**

共用プロダクト・オブジェクト・ツリー (SPOT) - **/usr** ファイルシステムに相当。

##### **tmp**

クライアント **/tmp** ディレクトリーの親ディレクトリー。

##### **vg\_data**

ボリューム・グループの復元時に使用される構成ファイル。

-t Type (続き)

マシン・タイプ

**alternate\_master**

1 次 NIM マスターが正常に機能しなくなった場合にバックアップとして確保されるマシン。

**diskless** すべてのリモートのファイルシステムおよびリソース。

**dataless** ローカル・ペーजング、ダンプ。リモート */*、**/usr**。リモートまたはローカルのその他のもの。

**standalone**

ローカルのファイルシステムおよびリソース。

**master** NIM 環境をコントロールするマシン。

ネットワーク・タイプ

**tok** トークンリング・ネットワーク。

**ent** イーサネット・ネットワーク。

**fddi** FDDI ネットワーク。

**atm** ATM ネットワーク。(AIX 4.3 以降。)

**generic** その他の TCP/IP ネットワーク。

グループ・タイプ

**mac\_group**

マシンのグループ。

**res\_group**

リソースのグループ。

## セキュリティ

アクセス制御: **nim** コマンドを実行するには、root 権限が必要です。

## 例

次に示す例は、操作ごとにまとめてあります。

### allocate

1. 名前が **syzygy** で **SPOT** 属性値が **spot1** であるディスクレス・ワークステーションにリソースを割り当てるには、次のように入力します。

```
nim -o allocate -a spot=spot1 syzygy
```

2. **krakatoa** というマシンで基本システムのインストールを実行するには、まず次のように入力して、リソースを割り当てる必要があります。

```
nim -o allocate -a spot=myspot -a lpp_source=images krakatoa
```

それから、次のように入力し、**bos\_inst** を実行してインストールをサポートするように NIM 環境を初期化します。

```
nim -o bos_inst krakatoa
```

3. インストール可能オプション **adt** が **lpp\_source** の **images** にある場合に、ソフトウェア・プロダクト **adt** をスタンドアロン・マシン **stand1** にインストールするには、次のように入力します。

```
nim -o allocate -a lpp_source=images stand1
```

それから、次のように入力します。

```
nim -o cust -a filesets="adt" stand1
```



4. インストール可能オプション `adt` のイメージが `lpp_source` の `images` にあり、`installp_bundle` の `bundle1` にインストール可能オプションの名前がある場合に、ソフトウェア・プロダクトをスタンドアロン・マシン `stand1` にインストールするには、次のように入力します。

```
nim -o allocate -a lpp_source=images \  
-a installp_bundle=bundle1 stand1
```

それから、次のように入力します。

```
nim -o cust stand1
```

5. BOS インストールのあとでネーム・レゾリューション・サービスを使ってマシンを自動的に構成するには、次のような内容でファイル `/exports/resolv.conf` を作成します。

```
nameserver      129.35.143.253  
nameserver      9.3.199.2  
domain          austin.ibm.com
```

それから、次のように入力します。

```
nim -o define -t resolv_conf -a location=/exports/resolv.conf \  
-a server=master rconf1
```

**bos\_inst** 操作を行う前に、次のように入力して、その他の必須リソースおよびオプション・リソースとともにこのリソースを割り当てます。

```
nim -o allocate -a spot=spot1 -a lpp_source=images1 \  
-a bosinst_data=bid1 -a resolv_conf=rconf1 client1
```

6. スタンドアロン・マシンに適用できるすべてのリソースを NIM リソース・グループ `res_grp1` からマシン `mac1` に割り当てるには、次のように入力します。

```
nim -o allocate -a group=res_grp1 mac1
```

## alt\_disk\_install

1. `hdisk4` および `hdisk5` で **image\_data** リソース `image_data_shrink` を使い、デバッグをオンにして、**mksysb** リソース `all_devices_msysb` をクライアント `roundrock` にインストールするには、次のように入力します。

```
nim -o alt_disk_install -a source=mksysb\  
-a image_data=image_data_shrink\  
-a debug=yes\  
-a disk='hdisk4 hdisk5' roundrock
```

2. (`alt_inst` ファイルシステムをマウントしたまま) `phase1` と `phase2` だけを実行させるために、クライアント `austin` 上の **rootvg** を `hdisk2` にクローンを作成するには、次のように入力します。

```
nim -o alt_disk_install -a source=rootvg\  
-a disk='hdisk2'  
-a phase=12 austin
```

## bos\_inst

1. リソース `spot1`、`images1`、`bosinst_data1`、および `rconf1` を使用してマシン `blowfish` をインストールするには、まず次のように入力して、リソースを割り当てます。

```
nim -o allocate -a spot=spot1 -a lpp_source=images1 \  
-a bosinst_data=bosinst_data1 -a resolv_conf=rconf1 blowfish
```

それから、次のように入力して、BOS インストールを実行します。

```
nim -o bos_inst blowfish
```

2. **bos\_inst** 操作を開始したら、リソース `spot1`、`images1`、`bosinst_data1`、および `rconf1` の割り当て中に自動的にマシン `blowfish` がインストールされるようにするには、次のように入力します。

```
nim -o bos_inst -a spot=spot1 -a lpp_source=images1 \  
-a bosinst_data=bosinst_data1 -a resolv_conf=rconf1 blowfish
```

3. マシン `mac1` のインストール時にデフォルト・リソースを使用するには、次のように入力します。

```
nim -o bos_inst mac1
```

4. `spot1` および `lpp_source1` を使用してマシン `deadfish` をインストールし、**adapter\_def** リソース `adapter_def1` を使用して 2 次アダプターを構成するには、次のように入力します。

```
nim -o bos_inst -a spot=spot1 -a lpp_source=lpp_source1 \  
-a adapter_def=adapter_def1 deadfish
```

5. マシン `blowfish` をインストールし、ソフトウェア・ライセンス契約を受け入れるには、次のように入力します。

```
nim -o bos_inst -a spot=spot1 -a lpp_source=images1 \  
-a accept_licenses=yes -a resolv_conf=rconf1 blowfish
```

## change

1. BLDG905 ネットワーク上のマシンは、`gateway905` ゲートウェイを使用して OZ ネットワークに到達します。OZ ネットワーク上のマシンは、`gatewayOZ` ゲートウェイを使用して BLDG905 ネットワークに到達します。BLDG905 および OZ という 2 つのネットワーク間に経路を追加するには、次のように入力します。

```
nim -o change -a routing1="OZ gateway905 gatewayOZ" BLDG905
```

2. ホスト名 `sailfish2.austin.ibm.com` で識別されるアダプターを、トークンリング・ネットワークに付加します。NIM マスターでこのアダプターの 2 次インターフェースを定義し、追加のイーサネット・ネットワークを表す NIM ネットワークを位置指定するよう指定し、それが見つからない場合は、サブネット・マスク `255.255.255.128` を使用して NIM ネットワークを定義するよう指定するには、次のように入力します。

```
nim -o change -a if2="find_net sailfish2.austin.ibm.com 0" \  
-a net_definition="tok 255.255.255.128" -a ring_speed2=16 master
```

**注:** ネットワークのデフォルト名が生成されますが、新規ネットワークのルーティング情報は指定されません。

3. それぞれデフォルト・ゲートウェイ `gw1` および `gw2` を使用するネットワーク `net1` および `net2` のデフォルト経路を定義するには、次の 2 つのコマンドを入力します。

```
nim -o change -a routing1="default gw1" net1  
nim -o change -a routing1="default gw2" net2
```

4. リソース・グループ `res_grp1` で定義されているリソース (そのリソースを適用できるすべての操作に対して常にデフォルトで割り当てられるリソース) を指定するには、次のように入力します。

```
nim -o change -a default_res=res_grp1 master
```

## check

1. `myspot` という SPOT の使用可能度について NIM 検査を行うには、次のように入力します。

```
nim -o check myspot
```

2. `images` という **lpp\_source** の状況を検査するには、次のように入力します。

```
nim -o check images
```

## cust

1. インストール可能オプション `adt` のイメージが **lpp\_source** の `images` にある場合に、ソフトウェア・プロダクトをスポット `spot1` にインストールするには、次のように入力します。

```
nim -o cust -a lpp_source=images -a filesets=adt spot1
```

2. インストール可能オプション **adt** のイメージが **lpp\_source** の **images** にあり、**installp\_bundle** の、**bundle1** にインストール可能オプションの名前がある場合に、ソフトウェア・プロダクトをスポット **spot1** にインストールするには、次のように入力します。

```
nim -o cust -a lpp_source=images -a installp_bundle=bundle1 spot1
```

3. **spot1** のインストール可能オプション **adt** のイメージが、このスポットのあるマシンにとってローカルなテープ・ドライブ内のテープにある場合、このスポットにソフトウェア・プロダクトをインストールするには次のように入力します。

```
nim -o cust -a lpp_source=/dev/rmt0 -a filesets=adt spot1
```

4. **spot1** のインストール可能オプション **adt** のイメージが、このスポットのあるマシンにとってローカルなテープ・ドライブ内のテープにある場合、このスポットにソフトウェア・プロダクトをインストールするには次のように入力します。

```
nim -o cust -a lpp_source=/dev/rmt0 -a filesets=adt spot1
```

5. APAR **IX12345** に関連し、テープ **/dev/rmt0** に常駐するすべてのファイルセット更新を、**spot1** および **spot1** に現在割り当てられているすべてのディスクレス・クライアントまたはデータレス・クライアントにインストールするには、次のように入力します。

```
nim -F -o cust -afixes=IX12345 -a lpp_source=/dev/rmt0 spot1
```

6. クライアント **Standalone1** にインストールされたすべてのソフトウェアを、**updt\_images** という **lpp\_source** 内の最新の更新で更新するには、次のように入力します。

```
nim -o allocate -a lpp_source=updt_images Standalone1  
nim -o cust -afixes=update_all Standalone1
```

7. マシン **catfish** を **installp\_bundle** の **bundle1** の内容とともにインストールするには、まず次のように入力して、リソースを割り当てます。

```
nim -o allocate -a installp_bundle=bundle1 \  
-a lpp_source=images1 catfish
```

それから、次のように入力して、**cust** 操作を行います。

```
nim -o cust catfish
```

8. クライアント **Standalone1** にインストールされたすべてのソフトウェアを、**updt\_images** という **lpp\_source** 内の最新の更新で更新するには、次のように入力します。

```
nim -o cust -a lpp_source=updt_images -a fixes=update_all \  
Standalone1
```

9. **cust** 操作を実行したら、リソースおよび **lpp\_source** **images1** の割り当て中にマシン **catfish** が **installp\_bundle** **bundle1** の内容とともにインストールされるようにするには、次のように入力します。

```
nim -o cust -a installp_bundle=bundle1 -a lpp_source=images1 \  
catfish
```

10. **adaper\_def** リソース **adapter\_def1** 内の 2 次アダプター構成ファイルを使用してクライアント・マシン **deadfish** 上に 2 次アダプターを構成するには、次のように入力します。

```
nim -o cust -a adapter_def=adapter_def1 deadfish
```

## deallocate

**images** という名前の **lpp\_source** をスタンドアロン・マシン **client1** から割り当て解除するには、次のように入力します。

```
nim -o deallocate -a lpp_source=images client1
```

## define

1. net1 というトークンリング・ネットワーク上で、AIX 5.1 バージョン 5.1 またはそれ以前の rspc ユニプロセッサ・データレス・マシンを定義し、これを altoid と呼ぶには、次のように入力します。

```
nim -o define -t dataless -a if1="net1 fred 10005aa88500" \  
-a ring_speed=16 -a platform=rspc -a netboot_kernel=up \  
-a comments="Dataless client altoid"
```

comments 属性はオプションであり、ユーザー入力の記述を入れることができます。

注: if1 属性は必須です。

2. サーバー altoid に位置し、パス名 /usr/sys/inst.images を持つインストール可能イメージを含むディレクトリであるリソースを定義して、そのリソースに images という名前を付けるには、次のように入力します。

```
nim -o define -t lpp_source -a server=altoid \  
-a location=/usr/sys/inst.images images
```

3. images という名前の 1 つの lpp\_source を使用し、/export/exec directory の NIM の NIM マスターに myspot という名前の新しい SPOT を作成するには、次のように入力します。

```
nim -o define -t spot -a server=master -a location=/export/exec \  
-a source=images myspot
```

4. サブネット・マスク 255.255.240.0 およびアドレス 129.35.129.0 を使用して、BLDG905 という名前のネットワーク・オブジェクトを定義するには、次のように入力します。

```
nim -o define -t tok -a snm=255.255.240.0 \  
-a net_addr=129.35.129.0 BLDG905
```

5. テープ上にある特定のソフトウェア・プロダクト bos.INed および bos.adt を選択して、テープからマスターに位置指定するような lppsrc1 を lpp\_source として定義するには、次のように入力します。

```
nim -o define -t lpp_source -a location=/images2/lppsrc1 \  
-a source=/dev/rmt0 -a server=master -a packages="bos.INed \  
bos.adt" lppsrc1
```

6. BOOTP 可能 IPL ROM 搭載の AIX 5.1 またはそれ以前のバージョン用の対称型マルチプロセッサ・モデルであるスタンドアロン・マシンを定義するには、次のように入力します。ただし、ホスト名は jupiter であり、NIM 環境でも jupiter として認識され、また net1 と呼ばれるトークンリング・ネットワークおよびリング速度 16 を使用するものとします。

```
nim -o define -t standalone -a if1="net1 jupiter 0" \  
-a ring_speed=16 -a platform=rs6ksmp jupiter
```

7. マスターの /resources/mksysb.image にある既存 mksysb イメージから、mksysb リソース、mksysb1 を定義するには、次のように入力します。

```
nim -o define -t mksysb -a server=master \  
-a location=/resources/mksysb.image mksysb1
```

8. ATM ネットワークを表すために、サブネット・マスク 255.255.240 およびアドレス 129.35.101.0 を使用して、ATMnet という名前の NIM ネットワークを定義するには、次のように generic ネットワーク・タイプを使用します。

```
nim -o define -t generic -a snm=255.255.240.0 \  
-a net_addr=129.35.101.0 ATMnet
```

9. AIX 5.1 またはそれ以前用の、ホスト名が bluefish である PowerPC PCI バス・ベースの対称型マルチプロセッサ・コンピューターを、net1 と呼ばれるトークンリング・ネットワーク上のスタンドアロン・マシンとして定義し、そのマシンを bluefish として NIM 環境に認識させるには、次のように入力します。

```
nim -o define -t standalone -a platform=rspcsmp \  
-a if1="net1 bluefish 0" -a ring_speed=16 bluefish
```

10. Disk1sMac1 という名前のマシン・グループを disk1s1、disk1s2、および disk1s3 という NIM ディスクレス・マシンとなるメンバーで定義するには、次のように入力します。

```
nim -o define -t mac_group -a add_member=disk1s1 \  
-a add_member=disk1s2 -a add_member=disk1s3 Disk1sMac1
```

11. Disk1sRes1 というリソース・グループをリソース spot1、root1、dump1、paging1、home1、tmp1 で定義するには、次のように入力します。

```
nim -o define -t res_group -a spot=spot1 -a root=root1 \  
-a dump=dump1 -a paging=paging1 -a home=home1 -a tmp=tmp1 \  
Disk1sRes1
```

12. リソースの定義中に、**mksysb** リソース mksysb2 の定義に必要なスペースを表示し、クライアント client1 の mksysb イメージを作成して、そのイメージをマスターの /resources/mksysb.image に配置するには、次のように入力します。

**注:** このアクションで表示されるのは、mksysb 操作に必要なスペースだけです。そうでなければ、リソース作成は行われません。

```
nim -o define -t mksysb -a server=master \  
-a location=/resources/mksysb.image -a source=client1 \  
-a mk_image=yes -a size_preview=yes mksysb2
```

13. **mksysb** リソース mksysb2 を定義し、リソースの定義中に、クライアント client1 の mksysb イメージを作成して、そのイメージをマスターの /resources/mksysb.image に配置するには、次のように入力します。

```
nim -o define -t mksysb -a server=master \  
-a location=/resources/mksysb.image -a source=client1 \  
-a mk_image=yes mksysb2
```

14. **mksysb** リソース mksysb2 を定義し、リソースの定義中に、クライアント client1 の mksysb イメージを作成して、**-em** の mksysb フラグを使用して作成したそのイメージをマスターの /resources/mksysb.image に配置するには、次のように入力します。

```
nim -o define -t mksysb -a server=master \  
-a location=/resources/mksysb.image -a source=client1 \  
-a mk_image=yes -a mksysb_flags=em mksysb2
```

15. マスターの /resources/exclude\_file1 に配置される **exclude\_files** リソース exclude\_file1 を定義するには、次のように入力します。

```
nim -o define -t exclude_files -a server=master \  
-a location=/resources/exclude_file1 exclude_file1
```

16. ホスト名が redfish\_t.lab.austin.ibm.com である redfish というマシンには、リング・スピード 16 メガビットでトークンリング・ネットワークに接続する 1 次インターフェースがあります。redfish を、NIM 環境のスタンドアロン・マシンとして定義し、マシンの 1 次インターフェースが接続しているネットワークの名前を突き止めるよう NIM に命令するには、次のように入力します。

```
nim -o define -t standalone -a if1="find_net \  
redfish_t.lab.austin.ibm.com 0" -a ring_speed=16 redfish
```

17. ホスト名が bluefish\_e.lab.austin.ibm.com である >bluefish というマシンには、**bnc** の **cable\_type** でイーサネット・ネットワークに接続する 1 次インターフェースがあります。bluefish を NIM 環境のディスクレス・マシンとして定義し、マシンの 1 次インターフェースが接続しているネットワークの名前を突き止めるよう NIM に命令し、それが見つからない場合は名前 ent\_net、サブネット・マスク 255.255.255.128、およびホスト名 lab\_gate のゲートウェイを使用するデフォルト経路を使って NIM 環境を定義するには、次のように入力します。

```
nim -o define -t diskless -a if1="find_net \  
bluefish_e.lab.austin.ibm.com 0" -a net_definition="ent \  
255.255.255.128 lab_gate 0 ent_net" -a cable_type=bnc bluefish
```

注: マスターのデフォルト経路が既に存在する場合は **net\_definition** 属性のマスター・ゲートウェイの位置で 0 を指定し、存在しない場合は、マスター・ゲートウェイを指定します。

18. **/export/nim/adapters** ディレクトリーを、マスターに配置される **adapter\_def** リソース **adapter\_def1** として定義するには、次のように入力します。

```
nim -o define -t adapter_def -a server=master \  
-a location=/export/nim/adapters adapter_def1
```

2 次アダプター構成ファイルを含んだ **adapter\_def** リソースを転送するには、**nimadapters** コマンドを実行します。

19. リソースの定義中に、**savevg** リソース **savevg2** の定義に必要なスペースを表示し、クライアント **client1** の **savevg** イメージを作成して、そのイメージをマスターの **/export/nim/savevg** に置き、バックアップをとる **volume\_group** が **myvg** となるようにするには、次のように入力します。

```
nim -o define -t savevg -a server=master \  
-a location=/export/nim/savevg/savevg2 -a source=client1 \  
-a mk_image=yes -a size_preview=yes -a volume_group=myvg savevg2
```

注: このアクションで表示されるのは、操作に必要なスペースだけです。 **savevg** またはリソース作成は行われません。

20. リソースの定義中に、**savevg** リソース **savevg2** を定義し、クライアント **client1** の **savevg** イメージを作成して、そのイメージをマスターの **/export/nim/savevg** に置き、バックアップをとる **volume\_group** が **myvg** となるようにするには、次のように入力します。

```
nim -o define -t savevg -a server=master \  
-a location=/export/nim/savevg -a source=client1 \  
-a mk_image=yes -a volume_group=myvg savevg2
```

21. **savevg** リソース **savevg2** を定義し、リソースの定義中に、クライアント **client1** の **savevg** イメージを作成して、**-em** の **savevg** フラグを使用して作成したそのイメージをマスターの **/export/nim/savevg** に配置するには、次のように入力します。

```
nim -o define -t savevg -a server=master \  
-a location=/export/nim/savevg -a source=client1 \  
-a mk_image=yes -a volume_group=myvg -a savevg_flags=em savevg2
```

22. ロケーション **/export/nim** にあるマスターに配置される **vg\_data** リソース **my\_vg\_data** を定義するには、次のように入力します。

```
nim -o define -t vg_data -a server=master -a location=/export/nim/my_vg_data my_vg_data
```

## dkls\_init

1. リソース **spot1**、**root1**、**dump1**、および **paging1** を使用して、**syzygy** という名前のディスクレス・ワークステーションの環境を初期化するには、まず次のように入力して、リソースを割り当てます。

```
nim -o allocate -a spot=spot1 -a root=root1 -a dump=dump1 \  
-a paging=paging1 syzygy
```

それから、次のように入力して、クライアント・マシンのリソースを初期化します。

```
nim -o dkls_init syzygy
```

2. **syzygy** という名前のディスクレス・ワークステーションの環境を初期化するには、次のように入力します。

```
nim -o dkls_init syzygy
```



3. `diskls2` というメンバーをマシン・グループ `DisklsMac1` 上の操作から除外し、`DisklsRes1` というリソース・グループで定義されたディスクレス・リソースの割り当て中に残りのメンバーを初期化するには、次の 2 つのコマンドを入力します。

```
nim -o select -a exclude=diskls2 DisklsMac1
nim -o dkls_init -a group=DisklsRes1 DisklsMac1
```

4. `dkls_init` 操作を実行している場合に、リソース・グループ `dk_resgrp1` で定義されている必須およびオプションのリソースの割り当て中に、マシン・グループ `dtgrp1` で定義されているディスクレス・マシンのグループを初期化するには、次のように入力します。

```
nim -o dkls_init -a group=dtgrp1 dk_resgrp1
```

## dtls\_init

1. リソース `spot1`、`root1`、および `dump1` を使用して、`syzygy` という名前のデータレス・ワークステーションの環境を初期化するには、まず次のように入力して、リソースを割り当てます。

```
nim -o allocate -a spot=spot1 -a root=root1 -a dump=dump1 syzygy
```

それから、次のように入力して、クライアント・マシンのリソースを初期化します。

```
nim -o dtls_init syzygy
```

2. `syzygy` という名前のデータレス・ワークステーションの環境を初期化するには、次のように入力します。

```
nim -o dtls_init syzygy
```

3. `dataless1` というメンバーをマシン・グループ `Data1sMac1` 上の操作から除外するには、`Data1sRes1` というリソース・グループで定義されたデータレス・リソースの割り当て中に残りのメンバーを初期化するには、次の 2 つのコマンドを入力します。

```
nim -o select -a exclude=dataless2 Data1sMac1
nim -o dtls_init -a group=Data1sMac1 Data1sRes1
```

4. `dtls_init` 操作を実行している場合に、リソース・グループ `Data1sRes1` で定義されている必須およびオプションのリソースの割り当て中に、マシン・グループ `Data1sMac1` で定義されているデータレス・マシンのグループを初期化するには、次のように入力します。

```
nim -o dtls_init -a group=Data1sMac1 Data1sRes1
```

## fix\_query

20 個の APAR 番号用にクライアント `Standalone1` にインストールした固定に関する情報をリストするには、次のように、行ごとに APAR 番号を使用して、ファイル `/tmp/apar.list` を作成します。

```
IX123435
IX54321
IX99999
...
```

それから、次のように入力します。

```
nim -o define -t fix_bundle -allocation=/tmp/apar.list \
    -aserver=master fix_bun
nim -o allocate -a fix_bundle=fix_bun Standalone1
nim -o fix_query Standalone1
```

## lppchk

1. `SPOT spot1` でファイルセット・バージョンおよび必要な整合性を検査するには、次のように入力します。

```
nim -o lppchk spot1
```

2. ファイル checksums で、スタンドアロン・マシン macgrp1 のグループ内にある NIM ターゲット上の、名前 bos で始まるすべてのパッケージを調べ、詳しいエラー情報を表示して、実際のファイル checksum との矛盾が見つかった場合にそれと一致するようソフトウェア・データベースを更新するには、次のように入力します。

```
nim -o lppchk -a lppchk_flags='-c -m3 -u' \  
-a filesets='bos*' macgrp1
```

**lppchk** 操作はデフォルトではグループ・メンバーのバックグラウンドで実行されるので、**lppchk** 操作の出力を表示するには、次のように入力します。

```
nim -o showlog -a log_type=lppchk macgrp1
```

## lppmgr

1. スペース使用情報を含んだ lpp\_source1 から除去する必要がある基本レベルの重複ファイルセットの名前をリストするには、次のように入力します。

```
nim -o lppmgr -a lppmgr_flags="-lsb" lpp_source1
```

2. 重複する基本ファイルセットと更新ファイルセットおよび取り替えられた更新を lpp\_source1 から除去するには、次のように入力します。

```
nim -o lppmgr -a lppmgr_flags="-rbux" lpp_source1
```

3. すべての非 SIMAGES (bos インストールに不要なファイルセット) を lpp\_source1 から除去するには、次のように入力します。

```
nim -o lppmgr -a lppmgr_flags="-rX" lpp_source1
```

4. C を除くすべての言語サポートを lpp\_source1 から除去するには、次のように入力します。

```
nim -o lppmgr -a lppmgr_flags="-r -k C" lpp_source1
```

## maint

1. ソフトウェア・プロダクト bos.INed および adt をスポット spot1 から撤去するには、次のように入力します。

```
nim -o maint -a installp_flags="-u" \  
-a filesets="bos.INed adt" spot1
```

2. **installp\_bundle** の bundle2 にインストール可能オプションの名前がある場合に、オプション bos.INed および adt をスポット spot1 から撤去するには、次のように入力します。

```
nim -o maint -a installp_flags="-u" \  
-a installp_bundle=bundle2 spot1
```

3. スポット spot1 で割り込まれたソフトウェア・インストールからクリーンアップするには、次のように入力します。

```
nim -o maint -a installp_flags="-C" spot1
```

4. マスターで、ソフトウェア・プロダクト bos.INed および adt を、スタンドアロン・マシン stand1 から撤去するには、次のように入力します。

```
nim -o maint -a installp_flags="-u" \  
-a filesets="bos.INed adt" stand1
```

5. マスターで、スタンドアロン・マシン stand1 から割り込まれたソフトウェア・インストールからクリーンアップするには、次のように入力します。

```
nim -o maint -a installp_flags="-C" stand1
```



6. マスターで、`bundle2` という **installp\_bundle** にインストール可能オプションの名前が入るように、ソフトウェア・プロダクト `bos.INed` および `adt` をスタンドアロン・マシン `stand1` からアンインストールするには、次のように入力します。

```
nim -o maint -a installp_flags="-u" \  
-a installp_bundle=bundle2 stand1
```

## **maint\_boot**

NIM スタンドアロン・クライアント `stand1` を保守モードでブートできるようにするには、次のように入力します。

```
nim -o maint_boot stand1
```

これによって保守ブート操作が設定されますが、ネットワーク・ブートは `stand1` からローカルで開始する必要があります。

## **remove**

`dump_files` という名前のリソースを除去するには、次のように入力します。

```
nim -o remove dump_files
```

## **showlog**

グループ `Disklsmacs1` で定義されているマシンのブート・ログを表示するには、次のように入力します。

```
nim -o showlog -a log_type=boot Disklsmacs1
```

## **showres**

1. `config` スクリプト `script1` の内容を表示するには、次のように入力します。

```
nim -o showres script1
```

2. `bosinst.data` リソース `bosinst_data1` の内容を表示するには、次のように入力します。

```
nim -o showres bosinst_data1
```

3. `lpp_source` `lpp_source1` のすべてのファイルセットをリストするには、次のように入力します。

```
nim -o showres lpp_source1
```

4. マシン `machine1` に現在インストールされているものに関連する `lpp_source` `lpp_source1` のすべてのファイルセットをリストするには、次のように入力します。

```
nim -o showres -a reference=machine1 lpp_source1
```

5. `lpp_source1` という `lpp_source` 上の `bos.INed` ファイルセットおよび `x1C.rte` ファイルセットに対するユーザー命令をリストするには、次のように入力します。

```
nim -o showres -a filesets="bos.INed x1C.rte" \  
-a installp_flags="qi" lpp_source1
```

6. `lpp_source1` という `lpp_source` 上のソフトウェアによって修正されたすべての問題をリストするには、次のように入力します。

```
nim -o showres -a instfix_flags="T" lpp_source1
```

7. クライアント `deadfish` に関する **adapter\_def** リソース `adapter_def1` 内の 2 次アダプター構成ファイルの内容を表示するには、次のように入力します。

```
nim -o showres -a client=deadfish adapter_def1
```

8. **adapter\_def** リソース `adapter_def1` 内のすべての 2 次アダプター構成ファイルの内容を表示するには、次のように入力します。

```
nim -o showres adapter_def1
```

9. **savevg** リソース `savevg1` の内容を表示するには、次のように入力します。

```
nim -o showres savevg1
```

## update

1. `/dev/cd0` 上のすべてのファイルセットを `lpp_source1` に追加するには、次のように入力します。

```
nim -o update -a packages=all -a source=/dev/cd0 lpp_source1
```

2. `bos.games 5.2.0.0` および `bos.terminfo` ファイルセットを `lpp_source1` に追加するには、次のように入力します。

```
nim -o update -a packages="bos.games 5.2.0.0 bos.terminfo" \  
-a source=/dev/cd0 lpp_source1
```

3. `bos.games` を `lpp_source1` から除去するには、次のように入力します。

```
nim -o update -a rm_images=yes -a packages="bos.games" lpp_source1
```

4. `lpp_source1` の欠落 `SIMAGES` を AIX インストール CD からリカバリーするには、次のように入力します。

```
nim -o update -a recover=yes -a source=/dev/cd0 lpp_source1
```

## ファイル

`/etc/niminfo` NIM が使用する変数が入っています。

## 関連情報

`lsnim` コマンド、`nimadapters` コマンド、`nimclient` コマンド、`nimconfig` コマンド、`niminit` コマンド。

`.info` ファイル。

---

## nim\_clients\_setup コマンド

### 目的

クライアントを定義し、NIM クライアント・オブジェクトに対する BOS インストール操作を初期化します。

### 構文

```
nim_clients_setup [ -m mksysb_resource] [ -n ] [ -c ] [ -r ] [ -v ] client_object(s)
```

### 説明

`nim_clients_setup` コマンドは、以下に示すタスクを実行することにより、NIM 環境において新しいクライアント・オブジェクトを定義し、クライアントのための BOS インストール操作を初期化します。

- 環境変数 `NIM_LICENSE_ACCEPT=yes` をエクスポートします。
  - ネットワーク・インストール中にソフトウェア使用条件を受け入れるために使用。
- `/etc/environment` に変数エントリ `NSORDER=local,bind` を追加します。
  - ホストが `/etc/host` の中にしか存在しない場合に、ネーム・レゾリューション用に必要。

- **client.defs** ファイルを使用してクライアント・オブジェクトを定義します (**-c** フラグが指定されている場合)。
  - ユーザーは、**nim\_clients\_setup** を使用する前に、**/export/nim/client.defs** ファイルの中のスタンザを編集する必要があります。
- クライアント・オブジェクトをインストール用に準備します。
  - **-c** フラグが使用されているなら、定義されるクライアントがインストール用に初期化されます。
  - クライアント・オブジェクトが指定されている場合、指定されたクライアントがインストール用に初期化されます。
  - **-c** またはクライアント・オブジェクトが指定されていない場合は、既存のすべての NIM クライアントがインストール用に初期化されます。
- BOS インストール操作では、グループ名 **basic\_res\_grp** に含まれるリソースがリソースとして使用されます。

注: **basic\_res\_grp** リソース・グループには、**nim\_master\_setup** コマンドの実行中に作成されたリソースが入れられます。このグループが存在しない場合は、**nim\_clients\_setup** コマンドを使用する前に、NIM インストール・リソースによってそれを定義しておく必要があります。

## フラグ

- m mksysb\_resource** BOS インストールにおいて復元する代替バックアップ・イメージを指定します。  
*mksysb\_resource* の値としては、新しい **mksysb** リソースの定義に使用される NIM オブジェクトの名前または絶対パス・ロケーションを指定できます。デフォルトで、**mksysb\_resource** リソースは、**basic\_res\_grp** NIM リソース・グループから割り当てられます。
- n** BOS インストールにおいて、ネイティブ (**rte**) インストールを有効にし、復元バックアップ・イメージ (**mksysb**) を無視します。デフォルトでは、BOS インストール中に **mksysb\_restore** が実行されます。
- c** **client.defs** ファイルに基づいてクライアント・オブジェクトを定義します。  
**/export/nim/client.defs** ファイルが存在していなければならず、有効なクライアント定義情報がなければなりません。 **client.defs** ファイルは、**nim\_master\_setup** コマンドの実行中に作成されます。そのファイルが存在しない場合には、ユーザーが **/usr/samples/nim/client.defs** の **client.defs** ファイルをコピーし、編集することができます。
- r** BOS インストール操作の後、クライアント・オブジェクトをリポートします。デフォルトでは、クライアントはリポートされません。インストール用にリソースが割り当てられ、クライアントは希望するタイミングでリポートできます。
- v** コマンド実行時の詳細デバッグ出力を有効にします。

## セキュリティ

アクセス制御: **nim\_clients\_setup** コマンドを実行するには、root 権限が必要です。

## 位置

**/usr/sbin/nim\_clients\_setup**

## 例

1. **/export/nim/client.defs** ファイルに基づいてクライアント・オブジェクトを定義し、新しく定義したクライアントを **basic\_res\_grp** のリソースを使用して BOS インストール用に初期化した後、クライアントをリポートしてインストールを開始するには、次のように入力します。

```
nim_clients_setup -c -r
```

2. 復元イメージとしてバックアップ・ファイル **/export/resource/NIM/530mach.sysb** を使用することにより、クライアント `client1` および `client2` を BOS インストール用に初期化するには、

```
nim_clients_setup -m /export/resource/NIM/530mach.sysb \ client1 client2
```

3. **basic\_res\_grp** リソース・グループのリソースを使用して、NIM 環境に含まれるすべてのクライアントをネイティブ (**rte**) BOS インストール用に初期化するには、

```
nim_clients_setup -n
```

## ファイル

**/etc/niminfo**

NIM が使用する変数が入っています。

## 関連情報

**nim\_master\_setup** コマンド、**nim\_update\_all** コマンド、**nim\_master\_recover** コマンド、**nim** コマンド、**nimconfig** コマンド、および **nimdef** コマンド。

---

## nim\_master\_recover コマンド

### 目的

ネットワーク・インストール管理 (NIM) データベースのバックアップを別のマシンに復元し、その変更を反映するようにデータベースを更新します。

### 構文

```
nim_master_recover [ -f mstr_fileset_dir]
```

```
[ -n nimdef_file]
```

```
[ -r nimdb_file]
```

```
[ -i mstr_interface]
```

```
[ -D ] [ -R ] [ -S ] [ -p ] [ -s ] [ -u ] [ -v ]
```

```
[ -N mstr_net_info [ -t net_def ]]
```

### 説明

**nim\_master\_recover** コマンドを使用することにより、バックアップ tar ファイルから NIM データベースを復元および更新できます。古いマスター上にある NIM データベースをバックアップするには、**smit nim\_backup\_db** コマンドを実行します。それにより、デフォルトで **/usr/objrepos/nimdb.backup** という名前の tar ファイルが作成されます。**nimdb.backup** が新しいマスターにコピーされたなら、そのファイルの絶対パスを指定した **-r** フラグを渡します。tar ファイルのパスが **/usr/objrepos/nimdb.backup** なら、**-r /usr/objrepos/nimdb.backup** を **nim\_master\_recover** スクリプトに渡します。

このスクリプトは、マスターの 1 次ネットワーク・インターフェースに基づいて NIM データベースに含まれるマスター定義を更新します。**-i** フラグには、マスターとして使用する 1 次インターフェースを指定します。**en0** を使用するには、**-i en0** を **nim\_master\_recover** スクリプトに渡します。

**注:** ネットワーク定義に静的経路が含まれているデータベースから復元する場合は、復元された NIM データベースが正しくない場合があります。 **nim\_master\_recover** コマンドを使用すると、新しいマスターの 1 次インストールを追加する前に、古いマスター定義のすべてのインターフェースが除去されます。 **nim\_master\_recover** コマンドを実行した後、**lsnim -lc networks** を実行することによって、経路指定情報が正しいことを確認してください。復元後のデータベースのすべての NIM ネットワーク定義に動的経路が含まれている場合には、そのような状況にはならないはずですが。

このスクリプトは、NIM データベースの復元および更新以外にも、オプションとしていくつかの機能を実行します。1 つは、**bos.sysmgmt** パッケージのロケーションを指定した **-f** フラグが渡された場合に、**bos.sysmgmt.nim.master** ファイルセットをインストールすることです。例えば、**bos.sysmgmt** パッケージが **/export/latest/install/ppc** ディレクトリーにある場合は、**nim\_master\_recover** スクリプトに **-f /export/latest/install/ppc** を渡します。

このスクリプトは、常に各クライアントをリセットします。 **-u** フラグが渡されるとこのスクリプトは、クライアントに割り振られていることがデータベースによって示されている NIM リソースをアンエクスポートしようとします。各クライアントの NIM マスターのホスト名は、その **/etc/niminfo** ファイルに格納されています。各クライアントの **niminfo** ファイルを更新するには、**-s** フラグを渡します。

**注:** NIM クライアントのうち、実行されていないもの、ネットワーク接続のないもの、新しいマスター **rhost** 権限を許可しないもの、または **bos.sysmgmt.nim.client 5.1.0.10** 以上のパッケージがないものについては、その **niminfo** は更新されません。 **niminfo** ファイルが更新されていないクライアントがあると、**nim\_master\_recover** スクリプトは、そのことを報告します。

**-n** フラグを使用して **nimdef** ファイルを指定すると、新しいクライアントを環境に追加できます。**nimdef** ファイルの詳細については、「AIX Installation Guide」を参照してください。

最後に、このスクリプトは、NIM データベースの中にリソースが存在するかどうかを調べます。存在しないリソースは、このスクリプトによって削除されます。例えば、新しいマスターが NIM サーバーと通信できない場合、そのサーバー上で定義されているリソースは NIM データベースから除去されます。 **-R** フラグを渡すと、スクリプトはリソースの検査をしません。

**注:** データベースがバックアップされたマスター上で定義済みのリソースは、そのデータベースがいったんリストアされると使用できなくなります。ただし、**nim\_master\_recover** の実行前にそれらのリソースが新しいマスターにコピーされていた場合はその限りではありません。

出力はすべて **/var/adm/ras/nim.recover** に記録されます。スクリプトの実行が完了したなら、ログにエラーが記録されていないかどうか確認してください。

**nim\_master\_recover** コマンドは、**-N** フラグを指定して呼び出された場合には異なる動作をします。これにより、マスターは、自身の **if1** 属性でホスト名、IP アドレス、および NIM ネットワークを変更させることができます。オプションで、**-t** フラグが **-N** フラグとともに提供された場合には、新規 NIM ネットワークを作成することができます。変更が実際に発生したら NIM 環境が正しく作動するように、マスターのネットワーク名またはアドレスが実際に変更される前に、これらのフラグを指定してコマンドを実行する必要があります。マスターの NIM 属性が変更されると、コマンドは、環境内に定義されている各スタンドアロン・クライアントの **/rhosts** および **/etc/niminfo** の更新を試みます。この試みが失敗したクライアントがあれば、その NIM マスター情報は手動で更新させなければなりません。また、スタンドアロン・クライアントが自身の NIM マスターのネットワーク名を変更させた後は、マスターが新しいネットワーク名で稼働中になるまで、いかなる NIM 操作も実行することができなくなります。

## フラグ

<b>-D</b>	復元後のデータベースから、クライアント定義をすべて削除します。
<b>-f</b> <i>directory</i>	インストールする <b>bos.sysmgt.nim.master</b> ファイルセットが含まれているディレクトリ。
<b>-i</b> <i>interface</i>	このコマンドを実行するマシンの 1 次ネットワーク・インターフェース。
<b>-n</b> <i>nimdef</i>	新しいマシンを定義するために使用されるオプションの <i>nimdef</i> 。
<b>-N</b> <i>mstr_net_info</i>	マスターの <i>if1</i> 属性を変更し、環境内に定義されている各スタンドアロン・クライアントを、マスターの新しいネットワーク情報を使用して更新しようと試みます。 <i>mstr_net_info</i> 変数は次のように構成されます。 " <i>nim_net_name</i> [ <i>hostname</i> ] [ <i>cable_type</i> ]" ここで、 <i>hostname</i> および <i>cable_type</i> はオプションです。
<b>-p</b>	スクリプトがマシンをリセットする前に、マシン状態を印刷します。
<b>-r</b> <i>nimdb.backup</i>	復元される NIM データベース・バックアップ tar ファイル。
<b>-R</b>	各リソースが存在するかどうかを調べません。このスクリプトのデフォルトの動作では、各リソースを調べ、存在しない場合にはデータベースからその定義を除去します。
<b>-S</b>	SPOT リソースを検査しません。スクリプトのデフォルトの動作では、すべての SPOT を検査して、インストールがサポートされる状態になるようにします。例えば、この検査によりブート・イメージが作成されます。
<b>-s</b>	各クライアント上の <b>niminfo</b> ファイルの更新を試みます。NIM クライアントのうち、実行されていないもの、ネットワーク接続のないもの、新しいマスター <i>rhost</i> 権限を許可しないもの、または <b>bos.sysmgt.nim.client 5.1.0.10</b> 以上のパッケージがインストールされていないものについては、その <b>niminfo</b> は更新されません。
<b>-t</b> <i>net_def</i>	マスターの IP アドレスが変わり、マスターを入れる既存の NIM ネットワークが存在しない場合は、新規 NIM ネットワークを作成します。このフラグは、合わせて <b>-N</b> フラグが指定されているときにのみ、有効です。 <i>net_def</i> 変数は次のように構成されます。 " <i>nim_net_name net_type net_addr net_snm default_route</i> " ここで、 <i>net_type</i> は <i>ent</i> 、 <i>tok</i> 、 <i>atm</i> 、あるいは <i>fddi</i> のいずれかです。
<b>-u</b>	復元データベースの中で割り振り済みとしてリストされるリソースをすべてアンエクスポートします。スクリプトのデフォルトの動作では、リソースの割り振りを解除することなく、NIM データベースから割り振りを削除します。
<b>-v</b>	コマンド実行時の詳細デバッグ出力を有効にします。

## 位置

**/usr/sbin/nim\_master\_recover**

## 終了状況

成功するとゼロ (0) を戻します。

## セキュリティ

アクセス制御: **nim\_master\_recover** コマンドを実行するには、root 権限が必要です。

## 例

1. **/export/nim/nimdb.backup** ファイルと、1 次インターフェース *en0* を使用して NIM マスターをリカバリーするには、次のように入力します。

```
nim_master_recover -r /usr/objrepos/nimdb.backup -i en0
```

2. NIM マスターのリカバリーの前に **/export/lpp\_source/installp/ppc** から **bos.sysmgt.nim.master** ファイルセットをインストールするには、次のように入力します。

```
nim_master_recover -f /export/lpp_source/installp/ppc \  
-r /usr/objrepos/nimdb.backup -i en0
```



3. 各リソースが存在するかどうかを検査したりブート・イメージの再構築のための SPOT を検査したりすることなく、NIM マスターのリカバリーを実行するには、次のように入力します。

```
nim_master_recover -R -S -r /usr/objrepos/nimdb.backup -i en0
```

4. NIM マスターのリカバリーを実行しながら、割り振り済みのリソースをアンエクスポートし、各クライアントのリセットの前にクライアントの状態を印刷するには、次のように入力します。

```
nim_master_recover -u -p -r /usr/objrepos/nimdb.backup -i en0
```

5. NIM マスターのリカバリーを実行し、各クライアントの **/etc/niminfo** ファイルを更新するには、次のように入力します。

```
nim_master_recover -s -r /usr/objrepos/nimdb.backup -i en0
```

6. NIM マスターのリカバリーを実行し、各クライアントをデータベースから削除し、新しいクライアントを **nimdef** ファイル **/export/nim/nimdef** から削除するには、次のように入力します。

```
nim_master_recover -D -n /export/nim/nimdef -r /usr/objrepos/nimdb.backup -i en0
```

7. マスターのホスト名を **newhost.domain.com** に変更し、**net2** という名前別の既存の NIM ネットワークへ移動し、しかし現行の **cable\_type** 属性を保持するためには、次のように入力します。

```
nim_master_recover -N "net2 newhost.domain.com"
```

8. マスターのホスト名を **newhost.domain.com** に変更し、マスターの **cable\_type** を **bnc** に変更し、**new\_nim\_net** という名前でアドレスが **192.168.1.0**、サブネット・マスクが **255.255.255.0**、デフォルト・ゲートウェイが **192.168.1.1** である新規 NIM イーサネット・ネットワークに移動するには、次のように入力します。

```
nim_master_recover -N "new_nim_net newhost.domain.com bnc" \  
-t "new_nim_net ent 192.168.1.0 255.255.255.0 192.168.1.1"
```

## ファイル

**/etc/niminfo**

NIM が使用する変数が入っています。

**/var/adm/ras/nim.recover**

コマンド実行結果のログ情報が入れます。

## 関連情報

**nim\_clients\_setup** コマンド、**nim\_update\_all** コマンド、**nim\_master\_setup** コマンド、**nim** コマンド、**nimconfig** コマンド、および **nimdef** コマンド。

---

## nim\_master\_setup コマンド

### 目的

ネットワーク・インストール管理 (NIM) マスター・ファイルセットを初期化し、NIM マスターを構成し、インストールに必要なリソースを作成します。

### 構文

```
nim_master_setup [ -a [ mk_resource={yes|no} ] [ file_system=fs_name ] [ volume_group=vg_name ] [ disk=disk_name ] [ device=device ] ] [ -B ] [ -F ] [ -L ] [ -v ]
```

## 説明

**nim\_master\_setup** コマンドは、NIM マスター・ファイルセットを初期化し、NIM 環境を構成します。初期化が完了すると、**nim\_master\_setup** コマンドは、以下のタスクを実行することによって NIM 環境を構成します。

- どのボリューム・グループおよびファイルシステムに NIM リソースが入れられるかを判別します。
- 必要な場合、ボリューム・グループおよびファイルシステムを作成します。
- マスターの NIM **mksysb** を作成します。
  - バックアップ・イメージ。
- NIM **lpp\_source** リソースを作成します。
  - プロダクト・イメージのソース。
- NIM SPOT リソースを作成します。
  - 共有プロダクト・オブジェクト・ツリー (SPOT) - **/usr** ファイルシステムに相当。
- NIM **bosinst\_data** リソースを作成します。
  - BOS インストール時に使用される **config** ファイル。
- NIM **resolv\_conf** リソースを作成します。
  - ネーム・サーバー構成ファイル。
- インストール中に使用されるデフォルトのリソース・グループを定義します。デフォルトのリソース・グループには、コマンド実行中に定義されるすべての NIM リソースが入れられます。
- サンプル **client.defs** 構成ファイルを、定義済みの NIM システム・システムにコピーします。
  - NIM 環境でクライアントを追加するために編集できるサンプル・ファイル。

## フラグ

**-a** 次の **attribute=value** の組を割り当てます。

### **mk\_resource={yes|no}**

NIM リソースを作成するかどうかを指定します。 **no** に設定した場合、コマンド実行時に NIM リソースは作成されません。デフォルト値は **yes** です。

### **file\_system=fs\_name**

NIM リソースを作成するための絶対パス・ロケーションを指定します。 **fs\_name** が存在しないなら、**vg\_name** から定義されているボリューム・グループの中に論理ボリュームが作成されます。 **fs\_name** のデフォルト値は **/export/nim** です。

### **volume\_group=vg\_name**

新しい論理ボリュームを作成するために使用されるボリューム・グループ名を指定します。 **vg\_name** が存在しない場合、**disk\_name** から定義される物理ボリューム (ディスク) を使用して、ボリューム・グループが作成されます。 **vg\_name** のデフォルト値は **rootvg** です。

### **disk=disk\_name**

**vg\_name** ボリューム・グループ作成時に、物理ボリュームを指定します。 **disk\_name** が指定されていないなら、次に使用可能な (空の) 物理ボリュームが使用されます。

### **device=device**

NIM マスター・ファイルセットのインストールおよびリソースの作成において使用されるインストール・イメージの絶対パス・ロケーションを指定します。 **device** のデフォルト値は **/dev/cd0** です。

**-B** バックアップ・イメージの作成機能をオフにします。

**-F** ファイルシステムの作成を使用不可にします。



- L **lpp\_source** リソースの作成を使用不可にします。
- v コマンド実行時の詳細デバッグ出力を有効にします。

## 位置

`/usr/sbin/nim_master_setup`

## 終了状況

成功するとゼロ (0) を戻します。

## セキュリティ

アクセス制御: `nim_master_setup` コマンドを実行するには、`root` 権限が必要です。

## 例

1. NIM マスター・ファイルセットをインストールし、`/dev/cd1` にあるインストール・メディアを使用して NIM 環境を初期化するには、次のようにします。

```
nim_master_setup -a device=/dev/cd1
```

2. NIM インストール・リソースを作成することなく NIM 環境を初期化するには、次のように入力します。

```
nim_master_setup -a mk_resource=no
```

3. NIM 環境を初期化し、バックアップ・イメージを作成することなく NIM インストール・リソースを作成し、マウント・ポイント `/cdrom` にあるインストール・メディアを使用するには、次のように入力します。

```
nim_master_setup -a device=/cdrom -B
```

4. 既存の NIM 環境に含まれる NIM リソースを定義し、`/dev/cd0` にあるインストール・メディアを使用し、ボリューム・グループ `nimvg` の下に `/export/resources/NIM` という新しいファイルシステムを作成するには、次のように入力します。

```
nim_master_setup -a volume_group=nimvg \  
-a file_system=/export/resources/NIM
```

注: ファイルシステム `/export/resources/NIM` がその時点で存在しないなら、ボリューム・グループ `nimvg` の下にそれが作成されます。 `nimvg` ボリューム・グループが存在しないなら、ディスク属性が指定されていなかったため、次の空の物理ボリューム (ディスク) を使用してそれが作成されます。

## ファイル

`/etc/niminfo` NIM が使用する変数が入っています。  
`/var/adm/ras/nim.setup` コマンド実行結果のログ情報が入られます。

## 関連情報

`nim_clients_setup` コマンド、`nim_update_all` コマンド、`nim_master_recover` コマンド、`nim` コマンド、`nimconfig` コマンド、および `nimdef` コマンド。

---

## nim\_move\_up コマンド

### 目的

AIX 環境において新しいハードウェア (すなわち、POWER5 またはそれよりも新しいサーバー) が容易に使用できるようにします。

### 構文

```
nim_move_up { [ -S ] | [ -K [ -h control_host ] ] | [ -r [ -R ] [ -u ] ] } | { [ -c NIM_client ] [ -i target_ip [ -ending_ip ] ] [ -s subnet_mask ] [ -g gateway ] [ -h control_host ] [ -m managed_sys ] [ -V vio_server [ -e ] [ -D ] ] [ -l img_src ] [ -l resource_dir ] [ -t seconds ] [ -p loops ] [ -j nimadm_vg ] [ -L lpp_source ] [ -U spot ] [ -B bosinst_data ] [ -E exclude_files ] [ -C script_resource ] [ -b install_bundle ] [ -f fix_bundle ] { [ -n ] [ -d ] | -O } [ -q ] [ -T ] }
```

### 説明

**nim\_move\_up** コマンドにより、既存の AIX 環境のユーザーは新しいハードウェア (すなわち、POWER5 サーバーまたはそれよりも新しいもの) 上で使用できる諸機能が利用可能となります。このコマンドは、既存の AIX システムを POWER5 サーバー (またはそれよりも新しいもの) に置かれている LPAR に移行させるインターフェースを提供します。元のマシン上の AIX のレベルは、POWER5 ハードウェア上の操作をサポートするレベルにまで高められます。元のシステムのハードウェア・リソースは、同等の POWER5 ハードウェア上で厳密に複製されます。移行の終了とともに、同じシステムが POWER5 LPAR 上で完全に実行します。

さらに、**nim\_move\_up** では POWER5 サーバーのバーチャル I/O 機能が使用できます。これを行うにはオプションで、仮想化されたハードウェア (仮想ディスクや仮想イーサネットなど) にクライアントをマイグレーションします。

**nim\_move\_up** コマンドは、ネットワーク上の NIM クライアントのリモートからの管理とインストールを可能にする NIM 機能と NIM マスター機能に依存します。**nim\_move\_up** コマンドでは NIM マスターとその **nimadm** コマンドの使用を試行して、既存の NIM クライアント上の以下のアクションを完了させます。

1. クライアントのシステム・バックアップを作成する。
2. AIX のバックアップのレベルを移行する。
3. POWER5 サーバーに置かれた LPAR にバックアップをインストールする。このサーバーは、NIM 環境内では新しいスタンドアロン・クライアントとして表されます。

POWER5 ハードウェアのインストール前には、NIM マスター (**nim\_move\_up** コマンドが実行される) と既存ハードウェア上の NIM クライアントを構成する必要があります (制約事項を参照)。このクライアントは移行の開始点であり、最終的には POWER5 LPAR となります。

移行が正常に終了した後の状態は、次のとおりです。

- NIM マスターはそのまま変わりありません。
- POWER5 サーバー上の LPAR は元の NIM クライアントに対応しており、これは NIM マスターによって制御されます。
- HMC は、SSH を介して NIM マスターと通信して POWER5 サーバー上の LPAR を制御します。
- 元の NIM クライアントは影響を受けないままの状態、依然として NIM マスターの制御下に置かれています。

移行全体は、元のクライアントの一部では必要とされた停止時間もなく行われます。このプロセスは、順番に実行して行く各フェーズの中で完了させることができます。これにより、このプロセスに対する一層の制御が可能となります。また、別の方法としてフェーズの全体を一気に実行可能であり、それによって処理時にユーザー応答のための介入は不要となります。このコマンドは、**bos.sysmgt.nim.master** ファイルセットの一部として納入され、その実行には有効な NIM 環境が必要となります。

## 必須フラグ

**-c** *NIM\_client*

NIM スタンドアロン・クライアント (スタンドアロン・オブジェクト・タイプ) または NIM マシン・グループ (*mac\_group* オブジェクト・タイプ) のいずれかを指定します。ここで指定されたクライアントはネットワークを使用して NIM マスターからアクセス可能である必要があります。このクライアントは、このクライアント上で NIM マスターがコマンドを実行可能にする必要があります。この引数の中に NIM マシン・グループを指定した場合は、このグループが同じ NIM ネットワークに置かれていることが必要です。このクライアントは、POWER5 サーバー上の同等の LPAR に移行されるターゲット・マシンです。デフォルト・ゲートウェイの IP アドレスを指定します。このアドレスを使用して、POWER5 サーバーへの移行後にクライアントが構成されます。

**-g** *gateway*

**-h** *control\_host*

POWER5 サーバーのハードウェア制御に使用される HMC のホスト名または IP アドレスを指定します。

**-i** *target\_ip[-ending\_ip]*

IP アドレスを指定します。このアドレスを使用して、移行済みの新しいクライアントが POWER5 サーバーにインストールされた後で、そのクライアントが構成されます。NIM マシン・グループを **-c** オプションに指定する場合は、ここで IP アドレスの範囲を指定する必要があり、そのアドレス範囲は移行予定のクライアント数に見合う必要があります。

**-l** *img\_src*

インストール・イメージのソースに至るパスを指定します。このソースを使用して移行とインストールに必要な NIM リソースを作成します。このパスは、デバイス (AIX プロダクト・メディアを使用する場合は **dev/cd0** など) またはインストール・イメージを含むファイルシステム上のある場所に至るパスとすることができます。インストール・イメージの中にある AIX のレベルは、AIX 5L バージョン 5.2 (5200-04 推奨メンテナンス・パッケージ適用) またはそれ以降、あるいは AIX 5.3 またはそれ以降である必要があります。

**-l** *resource\_dir*

ファイルシステム上のある場所に至るパスを指定します。このファイルシステムには、**nim\_move\_up** コマンドを使用して作成された新しい NIM リソースが含まれることとなります。 **-L** および **-U** オプションを介して既存のリソースが指定されていないければ、この場所には LPP\_Source とスポットを収容できるほどのスペースが必要となります。

**-m** *managed\_sys*

POWER5 サーバーに対応する、HMC で追跡されたとおりの、管理対象システムの名前を指定します。

**-s** *subnet\_mask*

サブネット・マスクを指定します。このマスクを使用して、POWER5 サーバーへの移行後にクライアントが構成されます。

## 実行と制御のフラグ

**-d**

バックグラウンドで **nim\_move\_up** を実行して、端末装置の制御を呼び出し側に戻します。 **nim\_move\_up** の進行状況は、**-S** フラグを使用して追跡することができます。

**-K**

指定された HMC に関する SSH キーを構成します。これにより、パスワード・プロンプトを必要としない、NIM マスターからのコマンドのリモート側での無人実行が可能となります。このフラグは、他のオプション (**-h** オプションを除く) と一緒に使用できません。

**-n**

**nim\_move\_up** 移行プロセスの次のフェーズのみを実行します。このフェーズが完了または失敗した時点で、**nim\_move\_up** コマンドは終了します。このフラグが無指定の場合、後続のフェーズがすべて実行されて、後続のフェーズがすべて実行されたか、フェーズの 1 つが失敗した時点で **nim\_move\_up** が終了します。

**-O**

指定された値のみを保管します。他のオプションを介して指定された値を保管してから、どのフェーズも実行せずに終了します。このフラグを、他の実行と制御のフラグと共に使用することはできません。

**-q**

抑止モードを指定します。出力は端末装置では表示されませんが (ただしログには保持されます)。 **nim\_move\_up** が **-d** フラグ指定で実行される場合、このフラグは無効です。

**-r**

**nim\_move\_up** の構成を解除します。これにより、保管済みオプションを含めた、すべての保管済みデータ、フェーズ固有のデータ、および現行フェーズ情報がリセットされます。この操作を実行する必要があるのは、新しいクライアントまたはクライアント・セットの移行処理が繰り返し開始される場合です。

**-R**

環境の構成を解除するほかに、**nim\_move\_up** によって作成された NIM リソースをすべて除去します。このフラグは **-r** オプションと共に使用されるだけです。

**-S**

現行フェーズまたは次に実行されるフェーズの状況を表示します。保管される値も全部が同様に表示されます。

**nim\_move\_up** コマンドは、情報の表示直後に終了します。このフラグを他のオプションと一緒に使用することはできません。

## オプション・フラグ

**-b** *installp\_bundle*

既存の **installp\_bundle** NIM リソースを指定します。このオプションを指定した場合に、このリソースのソフトウェアがフェーズ 10 (ポストインストール・カスタマイズ) で新たに移行された LPAR のそれぞれにインストールされます。

**-B** *bosinst\_data*

**nim\_move\_up** で使用される既存の **bosinst\_data** NIM リソースを指定して、新しいクライアントを POWER5 LPAR にインストールします。このオプションが無指定の場合、**nim\_move\_up** では、デフォルトの無人インストール値を使用して **bosinst\_data** リソースが生成されます。既存のスクリプト NIM リソースを指定します。このリソースは、このオプションが指定された場合に、移行されたすべての新しい LPAR 上のフェーズ 10 (ポストインストール・カスタマイズ) で **nim\_move\_up** が実行するものです。

**-D**

バーチャル I/O サーバー LPAR の指定時に POWER5 サーバー上で新しい LPAR を作成する段階で、仮想 SCSI アダプターの代わりに物理ストレージ・コントローラーを強制的に使用します。このフラグは、**-V** オプションとともに使用する場合にのみ有効です。

**-e**

バーチャル I/O サーバー LPAR の指定時に POWER5 サーバー上で新しい LPAR を作成する段階で、共用イーサネット・アダプターの代わりに物理ネットワーク・アダプターを強制的に使用します。このフラグは、**-V** オプションとともに使用する場合にのみ有効です。

**-E** *exclude\_files*

既存の **exclude\_files** NIM リソースを指定します。このリソースは、元のクライアントの **mksysb** を作成するために **nim\_move\_up** が使用するものです。このオプションが無指定の場合、**nim\_move\_up** により、**exclude\_files** リソースが生成されます。このリソースは、バックアップから **/tmp** の内容が除外されたものです。

**-f** *fix\_bundle*

既存の **fix\_bundle** NIM リソースを指定します。このオプションを指定した場合に、このリソースの APAR は、フェーズ 10 (ポストインストール・カスタマイズ) で新たに移行された LPAR のそれぞれにインストールされます。

**-j** *nimadm\_vg*

データをキャッシュ処理するための基本となる **nimadm** 呼び出しで使用されるボリューム・グループを指定します。このオプションを指定しない場合、デフォルト値は **rootvg** です。

**-L** *lpp\_source*

既存の **LPP\_Source** NIM リソースを指定します。このリソースの AIX レベルにターゲット・クライアントが移行されます。このオプションが無指定の場合、**nim\_move\_up** は、**-I** オプションを介して提供されたインストール・イメージ・ソースから新しい **LPP\_Source** の作成を試みます。

<b>-p</b> <i>loops</i>	リソースの使用状況を分析する段階で、ターゲット NIM クライアント上でシステム分析ツールの実行回数を指定します。最終のリソース使用データは、各実行ループから得られた値の平均となります。移行された LPAR の派生元となる同等の POWER5 リソースを決めるときには、このデータが考慮されます。このオプションを指定しない場合、デフォルトは 1 ループです。
<b>-t</b> <i>seconds</i>	各ループが実行される時間を秒数で指定します。このオプションを指定しない場合、デフォルトは 10 秒です。
<b>-T</b>	ユーザー定義のボリューム・グループを元のクライアントから新しくマイグレーションした LPAR へトランスポートします。
<b>-u</b>	<b>nim_move_up</b> が完全に <b>nim_move_up</b> マイグレーション全体を「ロールバック」できるようにします。これは <b>-r</b> フラグと一緒に使用しなければなりません。
<b>-U</b> <i>spot</i>	既存のスポット NIM リソースを指定します。このリソースは、クライアントの移行およびインストールで使用されます。このオプションが無指定の場合、 <b>-L</b> および <b>-I</b> オプションで指定した <i>lpp_source</i> NIM リソースから新しいスポットが作成されます。
<b>-V</b> <i>vio_server</i>	<b>-m</b> フラグで指定した POWER5 サーバー上にあるパーティクル I/O サーバーの LPAR 名を指定します。

## 終了状況

<b>0</b> ゼロ以外の値	正常終了。 エラーが発生しました。
--------------------	----------------------

## セキュリティ

root ユーザーのみがこのコマンドを実行できます。

## 例

- 最初のフェーズを実行して、必須オプションをすべて構成するには (**nim\_move\_up** が構成されていないことと実行されていないことが必要です)、次のように入力します。

```
nim_move_up -c client1 -i 192.168.1.100 -s 255.255.255.0 -g 192.168.1.1 -h hmc1.mydomain.com -m \
my-p5 -l /big/dir -I /dev/cd0 -n
```

- nim\_move\_up** コマンドの環境状態 (保管済みの全構成入力を含め、さらに次に実行されるフェーズがどれかを含めて) を表示するには、次のように入力します。

```
nim_move_up -S
```

- 保管済みのホスト名を新しい名前に変更して、出力を抑制しながら次のフェーズを実行するには、次のように入力します。

```
nim_move_up -h hmc2.mydomain.com -n -q
```

- 残る全フェーズをバックグラウンドで実行し、全ライセンスの受け入れ合意を保管して、フェーズの実行開始後にプロンプトを戻すようにするには、次のように入力します。

```
nim_move_up -Y -d
```

- nim\_move\_up** の構成を解除し、保管された全入力を破棄して、フェーズ 1 を実行するためにこのコマンドをリセットするには、次のように入力します。



`nim_move_up -r`

`nim_move_up` により事前作成された全 NIM リソースは、NIM 環境では影響を受けず、必要に応じて、`nim_move_up` で使用されて他のクライアントを移行することになります。

## 制約事項

`nim_move_up` アプリケーションの実行前には、以下の NIM マスター要件が満たされていなくてはなりません。

- AIX 5L バージョン 5.3 (5300-03 推奨メンテナンス・パッケージ適用) またはそれ以降を実行
- Perl 5.6 またはそれ以降
- OpenSSH (Linux Toolbox CD から)
- 環境内で AIX 4.3.3 アップデート またはそれ以降を実行する 1 つ以上のスタンドアロン NIM クライアント
- プロダクト・メディア・バージョン AIX 5L バージョン 5.2 (5200-04 推奨メンテナンス・パッケージ適用) またはそれ以降、あるいはプロダクト・メディア・バージョン AIX 5.3 またはそれ以降 (同等の LPP\_Source とスポット NIM リソースも使用する可能性があります)。

さらに、以下の前提条件を満たす必要があります。

- ターゲット・クライアントの同等 POWER5 構成をサポートする十分な条件を備えたハードウェア・リソース搭載の POWER5 サーバー。
- クライアントの移行に仮想リソースを使用する場合に、バーチャル I/O サーバーがインストールされ、構成されている。
- POWER5 サーバーを制御する HMC。電源オン、電源オフ、および LPAR 作成を行うのに十分な特権を併せ持つ。

必ずしも前記要件すべてが満足されていない場合、あるいは `root` 以外のユーザーがこのコマンドを実行した場合は、`nim_move_up` コマンドは正常に実行しません。

## 実装上の固有な条件

`nim_move_up` コマンドでは、フェーズ方式を使用して、既存のクライアントを同等の POWER5 LPAR に移行します。処理を構成するのは以下のフェーズです。

1. **NIM リソースを作成する。** 移行手順の実行に必要な NIM リソースがまだ存在していなければ、ここで作成します。
2. **移行前ソフトウェアを評価する。** どのソフトウェアをインストールし、どのソフトウェアが移行不能かの評価を、各ターゲット・クライアント上で行います。LPP\_Source から欠落したソフトウェアを、`nim_move_up` に提供されるインストール・イメージ (プロダクト・メディアなど) のソースから追加します。
3. **クライアント・ハードウェアと使用データを収集する。** 各ターゲット・クライアントのハードウェア・リソースに関するデータを収集します。また、これらハードウェア・リソースの平均的使用を、所定の時間の上で評価してみます。
4. **POWER5 リソースのアベイラビリティ・データを収集し、クライアント・リソース・データを変換する。** 指定される管理対象システムを検索して、使用可能なハードウェア・リソースを求めます。前のフェーズで収集されたデータを使用して、管理対象システムの使用可能なリソースを使用する同等 LPAR 構成を導き出します。バーチャル I/O サーバー LPAR が指定されていて連動する場合は、導出

されたクライアント LPAR が、物理入出力リソースの代わりにバーチャル I/O リソースを使用して作成されます。バーチャル I/O サーバー上には、必要に応じて、該当のアダプターと構成が作成されます。

5. **ターゲット・クライアントのシステム・バックアップを作成する。** NIM が各ターゲット・クライアントの **mksysb** を実行した後で、対応する **mksysb** NIM リソースが作成されます。
6. **各システム・バックアップを移行する。** **nim\_move\_up** で指定された NIM リソースを使用して、それぞれの **mksysb** リソースが、**nimadm** コマンドによって新しいレベルの AIX に移行されます。元の **mksysb** NIM リソースは保存され、新たに移行された **mksysb** リソース用に新しい **mksysb** NIM リソースが作成されます。
7. **NIM リソースを新しい LPAR に割り当てる。** フェーズ 4 で作成して新たに導出された LPAR ごとに、**nim\_move\_up** に提供されているネットワーク情報を使用して、NIM スタンドアロン・クライアント・オブジェクトが作成されます。該当の NIM リソースが割り当てられ、**bos\_inst** プル操作がそれぞれの NIM クライアント上で実行されます (NIM ではそのクライアントのブートを試行しません)。
8. **LPAR 上でインストールを開始する。** 制御ホスト (HMC) を使用して、それぞれの LPAR がリポートされ、インストールが開始されます。このフェーズの実行は、インストールの開始後に停止します (すなわち、インストールの進行状況はモニターされません)。
9. **移行後のソフトウェアを評価する。** それぞれのインストールの完了後に、移行が全体として正常に完了したかどうかの評価が行われ、移行時に検出したソフトウェア上の問題に関するレポートが作成されます。どのファイルセットの移行に失敗した場合も、そのファイルセットに対して報告されたエラーを手動で訂正する必要があります。
10. **ポストインストールをカスタマイズする。** 代替 LPP\_Source、ファイルセット・リスト、またはカスタマイズ・スクリプトが提供されている場合は、指定された値を使用して、カスタマイズした NIM 操作が各クライアント上で実行されます。これにより、追加ソフトウェア・アプリケーションまたは追加カスタマイズのオプション・インストールが可能となります。

NIM クライアントを同等の POWER5 LPAR に正しく移行させるために、上記の各フェーズ (フェーズ 10 を除く。このフェーズはオプション) のそれぞれを完全かつ正しく実行する必要があります。全フェーズが正常終了した場合は、移行済みの LPAR を表す NIM 環境内に新しい NIM クライアント・オブジェクトが存在することになり、インストール・リソースの **nim\_move\_up** ソースを介して提供された AIX のレベルを実行していることとなります。

**nim\_move\_up** の実行に必要なすべての前提条件が満たされた後で、**nim\_move\_up** コマンドは、2 つのフェーズ (構成とフェーズ実行) で実行されます。

## 構成

**nim\_move\_up** コマンドがそのフェーズを開始する前に、アプリケーションに入力が必要です。必須の入力には、移行対象の NIM クライアント・リスト、新しい移行済み LPAR の TCP/IP 構成情報、および POWER5 サーバー名が含まれます。必須の **nim\_move\_up** 構成オプションの全リストについては、必須フラグを参照してください (**nim\_move\_up\_config** SMIT メニューの中では、これらのフラグに \* (アスタリスク) が付けられています)。バーチャル I/O サーバーが指定されているかどうかといったオプション入力も、**nim\_move\_up** の動作と移行処理の最終結果に影響します (バーチャル I/O サーバーを指定した場合は、バーチャル I/O リソースが移行済み LPAR の作成に使用されます)。

SMIT インターフェースを介して必須とオプションの入力を転送するには、次のいずれかのコマンドを入力します。

```
smitty nim_move_up_config
```



または

```
smitty nim_move_up
```

次に、**Configure nim\_move\_up Input Values** オプションを選択します。

メニューでは、ご使用の環境の要件を反映するオプションを値と共に指定します。**nim\_move\_up** コマンドの **SMIT** インターフェースの詳細については、以下の **SMIT** の使用方法に関するセクションを参照してください。

必要な入力を使用して **nim\_move\_up** コマンドの環境を構成した後では、指定した値が、**nim\_move\_up** コマンドの後続の実行段階を通して記憶されます。この状態が続くのは、**nim\_move\_up** コマンド環境の構成が解除されるまでです。値はいつでも、**SMIT** メニュー・インターフェースを介して、あるいはコマンド行のフラグを使用して新しい値を指定することにより、変更することができます。コマンド行インターフェースは、**nim\_move\_up** コマンド環境の構成にも使用することができます。

#### 注:

コマンド行インターフェースの使用時は、デフォルトにより、このコマンドに構成値を指定した場合は常時、**nim\_move\_up** コマンドはフェーズの実行も試みます。このコマンドを直接呼び出すときには、フェーズの実行が妨げられないように、**-O** フラグを使用してください。

#### フェーズ実行

すべてが入力された後で、フェーズ 1 でフェーズ実行が開始され、これが順次に継続されます。あるフェーズでエラーが検出された場合、**nim\_move\_up** は、次回、このコマンドを実行する時点で、失敗したフェーズを実行しようとしています。**nim\_move\_up** が次のフェーズのみを開始するか、残りのすべてのフェーズを試行するように、任意選択で指定することができます。

**SMIT** インターフェースを介して **nim\_move\_up** フェーズを開始するには、次のいずれかのコマンドを入力します。

```
smitty nim_move_up_exec
```

または

```
smitty nim_move_up
```

次に、**Execute the nim\_move\_up Phases** オプションを選択します。 **Execute All Remaining Phases?** オプションに応答してから、**Enter** を押してください。フェーズは実行を開始します。

**nim\_move\_up** がコマンド行を使用して次のフェーズのみを実行するように指定するには、次のコマンドを入力します。

```
nim_move_up -n
```

**nim\_move\_up** が残りのすべてのフェーズを実行するように指定するには、次のコマンドを入力します。

```
nim_move_up
```

各フェーズを実行する他に、このコマンドでは、該当のフラグが指定されている場合に保管済みの構成オプションを変更することもできます。

## SMIT の使用法

**nim\_move\_up** **SMIT** メニューには、**nim\_move\_up** 高速パスを使用してアクセスすることができます。**nim\_move\_up** のルート・メニューを呼び出すには、次のコマンドを入力します。

smitty nim\_move\_up

ルート・メニューを介して、以下の SMIT 画面にアクセスすることができます。

### Display the Current Status of nim\_move\_up (nim\_move\_up の現在の状況の表示)

**-S** フラグ指定で **nim\_move\_up** を実行することと同じです。次に実行されるフェーズと保管済みの全オプションのリストが表示されます。

### Configure nim\_move\_up Input Values (nim\_move\_up 入力値の構成)

この画面を使用して、**nim\_move\_up** への必須とオプションの全入力構成を構成することができます。フィールドに入れられるすべての値は保管されて、**nim\_move\_up** の後続の実行段階と、この SMIT 画面の後続の使用段階を通して記憶されます。この画面をいつでも使用して、フェーズの実行後に、保管された値を変更することができます。

### Execute nim\_move\_up Phases (nim\_move\_up フェーズの実行)

**nim\_move\_up** フェーズを実行するために簡単なインターフェースを提供します。このフェーズの問題の対応策に応じて、フェーズを 1 つずつ、あるいは同時に全部を実行することができます。

### Configure SSH Keys on Target HMC (ターゲット HMC に関する SSH キーの構成)

リモート制御ホスト (HMC) に関する SSH キーをセットアップするために簡単なインターフェースを提供します。これはコマンド行の **-K** フラグを渡すことに相当する作業です。リモート制御ホストに関する SSH キーの構成により、NIM マスターからのコマンドのリモートからの無人実行が可能となります。これは、フェーズの全部 (一部のフェーズでは、このシステム上でコマンドをリモートで実行) を完了するために必要なことです。

### Unconfigure nim\_move\_up (nim\_move\_up の構成解除)

**nim\_move\_up** コマンドの環境の構成を解除するためにインターフェースを提供します。これにより、次に実行するフェーズを含めたすべての状態情報、一部のフェーズの実行結果として生成された保管済みデータ・ファイル、およびすべての保管済み入力値が除去されます。オプションとして、**nim\_move\_up** を介して作成されたすべての NIM リソースも同様に除去できます。この画面で行うことは、**-r** コマンド行オプションに相当する作業です。

## 拡張使用法: mig2p5 フレームワークの理解

**mig2p5** フレームワークは **/var/mig2p5** ディレクトリーから成り、これは **nim\_move\_up** にとっては後続の呼び出しと呼び出しの間でその状態を記憶する手段として使用されます。その存在と、**nim\_move\_up** による、その使用は、ユーザーにとっては完全に透過的です。したがって、**nim\_move\_up** によりディレクトリーが作成され、そのディレクトリーが存在しなければ、その値が初期化されます。**nim\_move\_up** が構成されない場合は、このディレクトリーが除去されます。このディレクトリーの内容は読み取りが容易であり、**nim\_move\_up** に伴う問題のトラブルシューティングに大いに役立つものです。このディレクトリーには各フェーズの中で生成されたログのすべてが含まれ、編集可能なファイルも含まれます。この編集可能ファイルは、コマンド行では不可能な方法で **nim\_move\_up** の動作に影響を及ぼします (コマンド行では不可能なこととは、**nim\_move\_up** が特定フェーズをその順序に従わずに強制的に実行することなど)。

以下には、**/var/mig2p5** ディレクトリーの中の各ファイルの目的と内容が説明されています。

#### config\_db

コマンド行の引数または **nim\_move\_up\_config** SMIT メニューを介して **nim\_move\_up** に渡される保管済み構成オプションのすべてを含みます。ファイルの中の各行の形式は次のとおりです。

*option\_name:value*

#### current\_phase

**nim\_move\_up** の次の呼び出し時に実行されるフェーズの番号を含みます。当フェーズの実行前

に、前のすべてのフェーズが正常に実行されていることを **nim\_move\_up** が確認します。この情報は、**mig2p5** フレームワークを使用してどこでも維持されます。

#### **global\_log**

**mig2p5** フレームワークが最後に初期化されてから実行された全フェーズの出力を含みます。

#### **client\_data/**

フェーズ 3 と 4 で **nim\_move\_up** が生成したファイルを含みます。これらのフェーズでは、元のクライアントのシステム・リソースと使用状況がモニターされ、定量化されて構成ファイルに入れます。POWER5 サーバー内の使用可能なリソースも、定量化されて対応するテキスト・ファイルに入れます。POWER5 サーバー上で新たに派生した LPAR のハードウェア・プロファイルを決定時に、これらのファイル内のデータがすべて、考慮されることとなります。

**nim\_move\_up** コマンドの内部使用に備えて、これらのファイルは機械可読データ・ファイルとなるように意図されています。これらのファイルを手動で変更したり、作成したりしないでください。

#### **phase#**

対応するフェーズに固有のデータを含みます。フェーズはその名前の中の番号 (#) で示されます。どのフェーズにもディレクトリー (例えば、**phase1/**、**phase2/** など) があります。

#### **phase#/log**

フェーズの実行時に表示される全出力を含みます。フェーズを複数回実行すると (エラーが修正された後など)、新しい出力がすべて、ファイル内に既に存在するテキストの後ろに付け加えられます。このログが役に立つのは、このフェーズに関連する失敗原因を失敗発生後に調べる場合です。

**global\_log** ファイルは、各フェーズのログ・ファイルすべてから構成され、そのファイルの中の全出力はそのファイルが元々表示された順序で配置されます。

#### **phase#/status**

このフェーズが最後に実行された時点で正常に終了したか、失敗したかを示します。このファイルは、後続のフェーズの実行の可否を決めるために **nim\_move\_up** が使用します。フェーズが実行できるのは、前のフェーズの **status** ファイルに文字列 **success** が含まれている場合だけです。そのフェーズが、最後に実行された時点でフェーズが失敗に終わらせる原因となったエラーを検出した場合は、この **status** ファイルは **failure** 文字列を含みます。

**pid** **nim\_move\_up** のバックグラウンドでの実行時の **nim\_move\_up** プロセス ID 番号を含みます。処理の終了時にこのファイルの終結処理が行われます。このファイルが存在し、プロセス ID が含まれている限り、**nim\_move\_up** はフェーズを実行できません。その理由は、**nim\_move\_up** の並行実行はサポートされていないためです。

ログ・ファイルと、**client\_data/** ディレクトリーの内容の場合を除きますが、**mig2p5** フレームワークを構成するファイル (**/var/mig2p5** 内にある) を読み取りおよび変更可能であり、それによって、**nim\_move\_up** がそのコマンド行と SMIT インターフェース経由では不可能なタスクを実行します。お客様が **mig2p5** 環境を操作することをお勧めします。それによって、**nim\_move\_up** を特定の要求に合わせて、移行処理時に起こる可能性のある問題のトラブルシューティングに役立つようにします。

**注:** **mig2p5** フレームワークをカスタマイズすることは、高度な使用法と考えられますので、正しくカスタマイズしないとその結果が不満足に終わる可能性があります。**mig2p5** 環境を直接、変更するユーザーは、行おうとする変更およびその変更が **nim\_move\_up** アプリケーションの動きに及ぼす影響に精通しているユーザーに限るようにしてください。

## ファイル

/usr/sbin/nim\_move\_up

nim\_move\_up コマンドが入っています。

## 関連情報

91 ページの『nim コマンド』、130 ページの『nimadm コマンド』。

---

## nim\_update\_all コマンド

### 目的

NIM リソースを更新し、NIM クライアントをカスタマイズします。

### 構文

```
nim_update_all [ -d device ] [ -l lpp_source resource ] [ -s spot resource ] [ -B ] [ -u ] [ -v ] client  
object(s)
```

### 説明

**nim\_update\_all** コマンドは、NIM 環境に含まれるインストール・リソースおよびクライアントを更新します。さまざまなフラグを使用することによって、どの NIM リソースを更新するかを指定したり、NIM クライアントの更新機能を使用不可にしたりできます。**nim\_update\_all** コマンドは、以下のタスクを実行することによって、NIM 環境を更新します。

- 環境変数 **NIM\_LICENSE\_ACCEPT=yes** をエクスポートします。
  - 更新インストール中にソフトウェア使用条件を受け入れるために使用。
- **/etc/environment** に変数エントリ **NSORDER=local,bind** を追加します。
  - ホストが **/etc/host** の中にしか存在しない場合に、ネーム・レゾリューション用に必要。
- メディアから更新レベル情報を取得します。
  - メディアのデフォルトのロケーションは、**/dev/cd0** です。
  - メディア・ロケーションは、**-d** フラグを使用することによって変更できます。
- **lpp\_source**、**spot**、および **mksysb** のリソースを更新します。
  - **lpp\_source** リソース名は、**-l** フラグを使用することによって指定できます。
  - **spot** リソース名は、**-s** フラグを使用することによって指定できます。
  - **mksysb** リソース名は、**basic\_res\_grp** リソース・グループに含まれる **mksysb** リソースから入手されます。**-B** フラグを指定すると、**mksysb** リソースの更新機能がオフになります。
- NIM クライアントに対して **update\_all** 操作を実行します。
  - クライアント・オブジェクトが指定されている場合、指定されたクライアントが更新されます。
  - クライアント・オブジェクトが指定されていない場合は、既存のすべての NIM クライアントが更新されます。
  - **-u** フラグが使用される場合、クライアントは更新されません。

## フラグ

<b>-d</b> <i>device</i>	このコマンドの実行中に使用する更新イメージの絶対パス・ロケーションを指定します。 <i>device</i> のデフォルト値は <b>/dev/cd0</b> です。
<b>-l</b> <i>lpp_source resource</i>	更新する <i>lpp_source resource</i> のオブジェクト名を指定します。このリソース名のデフォルト値は、 <b>basic_res_grp</b> から取得されます。
<b>-s</b> <i>spot resource</i>	更新する <i>spot resource</i> のオブジェクト名を指定します。このリソース名のデフォルト値は、 <b>basic_res_grp</b> から取得されます。
<b>-B</b>	<b>basic_res_grp</b> に含まれるバックアップ・イメージの更新を使用不可にします。
<b>-u</b>	クライアント・オブジェクトの更新機能を使用不可にします。
<b>-v</b>	コマンド実行時の詳細デバッグ出力を有効にします。セキュリティー

## 位置

**/usr/sbin/nim\_update\_all**

## 終了状況

成功するとゼロ (0) を戻します。

## セキュリティー

アクセス制御: **nim\_update\_all** コマンドを実行するには、**root** 権限が必要です。

## 例

- デバイス **/dev/cd2** にある更新イメージを使用して **basic\_res\_grp** リソース・グループに含まれるインストール・リソース **520lpp\_res** (**lpp\_source**)、**520spot\_res** (**spot**)、および **master\_sysb** (**mksysb**) を更新し、NIM 環境内のすべてのクライアントを更新するには、次のように入力します。  

```
nim_update_all -d /dev/cd2
```
- インストール・リソース **lpp1** (**lpp\_source**)、**spot1** (**spot**) を更新し、デバイス **/dev/cd0** にある更新イメージを使用して **mksysb** イメージの更新機能を使用不可にし、NIM 環境内のクライアント・オブジェクト **machine1** を更新するには、次のように入力します。  

```
nim_update_all -l lpp1 -s spot1 \  
-B machine1
```
- インストール・リソース **520lpp\_res** (**lpp\_source**)、**520spot\_res** (**spot**) を更新し、デバイス **/dev/cd0** にある更新イメージを使用することによって **basic\_res\_grp** リソース・グループに含まれる **mksysb** イメージの更新機能を使用不可にし、NIM 環境内のクライアントの更新機能を使用不可にするには、次のように入力します。  

```
nim_update_all -B -u
```

## ファイル

<b>/etc/niminfo</b>	NIM が使用する変数が入っています。
<b>/var/adm/ras/nim.update</b>	コマンド実行結果のログ情報が入れられます。

## 関連情報

**nim\_master\_setup** コマンド、**nim\_clients\_setup** コマンド、**nim\_master\_recover** コマンド、**nim** コマンド、および **nimconfig** コマンド。



---

## nimadapters コマンド

### 目的

ネットワーク・インストール管理 (NIM) の 2 次アダプター定義をスタンザ・ファイルから定義します。

### 構文

```
nimadapters {-p | -d | -r } -f SecondaryAdapterFileName adapter_def_name
```

または

```
nimadapters {-p | -d | -r } -a client=Client [-a info=AttributeList] adapter_def_name
```

### 説明

**nimadapters** コマンドは、NIM 2 次アダプター定義を *adapter\_def* リソースの一部として NIM 環境に追加するために必要なファイルを作成するために、2 次アダプター・スタンザ・ファイルを構文解析するためのものです。**nimadapters** コマンドは、2 次アダプターを構成しません。実際の構成は、**adapter\_def** リソースを参照する **nim -o bos\_inst** または **nim -o cust** 操作時に行われます。

**注:** **nimadapters** コマンドを使用する前に、NIM マスターを構成する必要があります。詳細については、インストールおよび移行の NIM マスターの構成と基本インストール・リソースの作成のセクションを参照してください。

### 2 次アダプター・ファイルの規則

2 次アダプター・ファイルのフォーマットには、次の規則があります。

- スタンザ・ヘッダーの後に、Attribute = Value 形式の属性行を続けます。
- 同じスタンザ内で、ある属性値を複数回定義すると、最後の定義だけが使用されます。
- 無効な属性キーワードを使用すると、その属性定義は無視されます。
- ファイルの各行に入れられるのは、1 つのヘッダーまたは属性定義だけです。
- 複数のスタンザを各マシン・ホスト名の定義ファイルに入れることができます。
- マシン・ホスト名の各スタンザは、その NIM クライアント上の 2 次アダプター定義を表します。同じマシン・ホスト名に対して、2 つの 2 次アダプター定義のロケーションまたは *interface\_name* を同じにすることはできません。特定の NIM クライアント上に存在できる定義は、アダプターまたはインターフェースごとに 1 つだけです。
- スタンザ・ヘッダー・エントリがキーワード **default** なら、そのスタンザを使用してデフォルト値を定義することを指定することになります。
- すべての 2 次アダプター属性に対してデフォルト値を指定できます。しかし、**netaddr** および **secondary\_hostname** 属性は固有でなければなりません。また、**location** および **interface\_name** は、NIM クライアント上で固有でなければなりません。
- 2 次アダプターの属性を指定せず、デフォルト値を定義すると、そのデフォルト値が使用されます。
- デフォルト値は、定義ファイル内のどのロケーションでも指定または変更できます。デフォルト値は、設定が済むと、それ以降のすべての定義に適用されます。
- あとに続くすべてのマシン定義に対するデフォルト値をオフにするには、デフォルト・スタンザで属性値を **nothing** に設定します。
- 単一のマシン定義に対するデフォルト値をオフにするには、マシン・スタンザで属性値を **nothing** に設定します。

- クライアント定義ファイルにはコメントを入れることができます。コメントは、# 文字で開始します。
- 定義ファイルの中でヘッダーや属性のキーワードおよび値の構文解析する際、タブ文字およびスペースは無視されます。

注: `nim -o bos_inst` または `nim -o cust` の操作時に、NIM がクライアント上の構成データを検査して、2 次アダプターが `adapter_def` リソースで要求される属性を使用して既に正確に構成されていると判断した場合、この 2 次アダプターは再構成されません。

## 2 次アダプター・ファイルのキーワード

2 次アダプター・ファイルでは、次のキーワードを使用してマシン属性を指定します。

### 必須属性:

#### **machine\_type = secondary | etherchannel | install**

`machine_type` 属性を `secondary` として指定すると、`nimadapters` 入力と `nimdef` 入力が明確に区別されます。`nimdef` コマンドに 2 次アダプター・ファイルが誤って渡された場合、エラーは容易に検出できます。`machine_type` が `install` であるスタンプは無視されます。

#### **netaddr**

2 次アダプターのネットワーク・アドレスを指定します。

#### **network\_type = en | et | sn | ml | vi**

ネットワーク・インターフェースのタイプを指定します。`en`、`et`、`sn`、`ml`、または `vi` のいずれかです。この属性は、推奨されない `network_type` 属性を置き換えます。

#### **subnet\_mask**

2 次アダプターが使用するサブネット・マスクを指定します。

### オプションの属性:

#### **adapter\_attributes**

物理アダプター属性と値を空白で区切ったリスト (例えば `"Attribute1=Value1 Attribute2=Value2"`)。要求された物理アダプターに対して設定できる属性のリストを表示するには、コマンド `lsattr -E -l AdapterName` を実行します。

#### **interface\_attributes**

インターフェースの属性と値を空白で区切ったリスト (例えば `"Attribute1=Value1 Attribute2=Value2"`)。要求されたインターフェースに対して設定できる属性のリストを表示するには、コマンド `lsattr -E -l InterfaceName` を実行します。この属性は、**attributes** 属性を置き換えます。

#### **cable\_type**

ケーブル・タイプを指定します (`network_type` が `en` または `et` の場合はオプションル)。

#### **comments**

2 次アダプター定義に入れるコメントを指定します。コメント文字列は二重引用符 (") で囲みます。

#### **interface\_name**

2 次アダプターのネットワーク・インターフェースの名前を指定します (例えば `en1`、`sn0`、`ml0` など)。 `location` と `interface_name` の両方を指定しないようにしてください。

注: `interface_name` は `interface_type` と整合性がなければなりません。

## location

このネットワーク・インターフェースに対応するアダプターの物理ロケーションを指定します。  
`location` と `interface_name` の両方を指定しないようにしてください。

**注:** 多重リンク疑似デバイスを除き、`location` を使用することを強くお勧めします。`location` を指定せずに、ユーザーが複数のアダプターを追加するかまたはオペレーティング・システムの再インストールと同時にアダプターを追加すると、アダプターおよびネットワーク・インターフェースの名前が、オペレーティング・システムによって予期しない方法で再割り当てされる可能性があります。

## multiple\_physloc

この属性は、イーサチャンネル・スタンザまたは VIPA スタンザとともに使用して、インターフェースに関連付ける物理アダプターを指定することができます。

## media\_speed

メディア・スピードを指定します (`network_type` が `en` または `et` の場合はオプション)。

## secondary\_hostname

`netaddr` 属性を使用して `/etc/hosts` ファイルに保管するホスト名。`hostname` コマンドまたは `uname -S` コマンドを使用した場合、このホスト名は設定されません。

## bos\_preconfig

`tunchange` がチューニング・パラメーターの値を変更することを指定します。`bos_preconfig` 属性を指定すると、`/usr/lpp/bos.sysmgmt/nim/methods/c_cfgadptrs` スクリプトでデフォルト値に設定されたチューナブル・パラメーターを変更できます。`bos_preconfig` 属性は `nim -o bos_inst` コマンドで使用されます。チューナブル値に関する有効なスタンザおよび推奨スタンザ・コマンドについて詳しくは、`tunchange` コマンドを参照してください。

`bos_preconfig` 属性のフォーマットは次のとおりです。

```
bos_preconfig="tunchange -f nextboot -t Stanza [ -o tunable=value ... ]"
```

**要件:** `bos_preconfig` 属性を使用して行った新しい設定を有効にするためには、システムを再始動する必要があります。

## cust\_preconfig

`vmo` コマンドがチューニング・パラメーターの値を変更することを指定します。`bos_preconfig` 属性を指定すると、`/usr/lpp/bos.sysmgmt/nim/methods/c_cfgadptrs` スクリプトでデフォルト値に設定されたチューナブル・パラメーターを変更できます。`cust_preconfig` 属性は `nim -o cust` コマンドで使用されます。有効なチューナブル・パラメーターについて詳しくは、`vmo` コマンドを参照してください。

`cust_preconfig` 属性のフォーマットは次のとおりです。

```
cust_preconfig="vmo -r [ -o tunable=value ... ]"
```

**注:** チューナブル・パラメーターを設定するには、`cust_preconfig` 属性を使用してシステムを再始動する必要があります。

**route** ネットワーク・ルーティング・テーブルに追加する経路の値を指定します。以下の値を指定します。指定の必要のない値は空白・スペースのままにします。

### 宛先 IP

送付先のホストまたはネットワーク。数字アドレスとして値を指定します。

### 宛先サブネット・マスク

宛先 IP が属するネットワークを決定するマスク。数字アドレスとして値を指定します。



## ゲートウェイ IP

パケットの送信先のネットワーク。数字アドレスとして値を指定します。

値はそれぞれ二重コロン (::) で区切り、さらに 3 つの値をセットとしてコンマ (,) で区切りま  
す。経路属性のフォーマットは次のとおりです。

```
route="DestHostA::MaskHostA::GatewayHostA, DestHostB::MaskHostB::GatewayHostB, ..."
```

適用されない値は空白にできますが、その場合も、次の例のように二重コロンで区切る必要が  
あります。

```
route="1.2.3.4:::5.6.7.8"
```

**nimadapters** コマンドで **-a info** フラグを使用して経路属性を追加する場合は、経路の値をそれ  
ぞれ二重コロンで区切り、さらに 3 つの値をセットとしてスペースで区切る必要があります。

## 2 次アダプター・ファイルのスタンザ・エラー

2 次アダプター・スタンザは、以下の条件のいずれかのもとではエラーを引き起こします。

- 定義のスタンザ・ヘッダーで使用されているホスト名が解決不能。
- 必須の属性が存在しない。
- 属性に無効な値を指定している。
- 属性ミスマッチが起こっている。例えば、`interface_type` が `en` または `et` でない場合、`cable_type=bnc` ま  
たは `media_speed=1000_Full_Duplex` を指定することはできません。
- スタンザに `location` 属性と `interface_name` 属性の両方が含まれている。
- 同じアダプター・ロケーションおよび同じホスト名に対して 2 次アダプター定義を複数回行っている。
- 同じ `interface_name` と同じホスト名に対して、2 次アダプターが複数回定義されている。

2 次アダプター・スタンザが正しくない場合は、エラーが報告され、そのスタンザは無視され、間違ってい  
るスタンザに関係なく次の入力処理されます。

**2 次アダプター・ファイルの例:** 以下は 2 次アダプター・ファイルの例を示したものです。

```
# Set default values.

default:

    machine_type = secondary

    subnet_mask  = 255.255.240.0

    network_type = en

    media_speed  = 100_Full_Duplex

# Define the machine "lab1"

# Take all defaults and specify 2 additional attributes.

# Unlike the case of the client definitions that are input to the

# nimdef command, the secondary adapter definition includes at least

# one required field that cannot be defaulted.

lab1:

    netaddr = 9.53.153.233
```

```

    location = P2-I1/E1

# Change the default "media_speed" attribute.

default:

    media_speed    = 100_Half_Duplex

# define the machine "test1"

# Take all defaults and include a comment.

test1:

    comments      = "This machine is a test machine."

# define a machine with a VIPA interface that uses interfaces en2 and en3.
lab2:
    machine_type      = secondary
    interface_type    = vi
    interface_name    = vi0
    netaddr           = 9.53.153.235
    subnet_mask       = 255.255.255.0
    secondary_hostname = lab3
    interface_attributes = "interface_names=en2,en3"

# define a machine with an etherchannel adapter that uses the adapters at
# the following location codes P1-I4/E1 and P1/E1
lab4:
    machine_type      = etherchannel
    interface_type    = en
    interface_name    = en2
    netaddr           = 9.53.153.237
    subnet_mask       = 255.255.255.0
    multiple_physloc  = P1-I4/E1,P1/E1

# define a machine with an etherchannel adapter that uses the
# ent2 and ent3 adapters and uses mode 8023ad.
lab6:
    machine_type      = etherchannel
    interface_type    = en
    interface_name    = en2
    netaddr           = 9.53.153.239
    subnet_mask       = 255.255.255.0
    adapter_attributes = "adapter_names=ent2,ent3 mode=8023ad"

```

## フラグ

**-a** 次の `attribute=value` の組を割り当てます。

### **client=nim\_client\_name**

2 次アダプター定義を追加または除去する NIM クライアントを指定します。このオプションを使用すると、1 つのクライアントに対して 1 つの 2 次アダプターを定義できます。複数の 2 次アダプターを定義するには、スタンザ・ファイルを使用します。

### **info=AttributeList**

2 次アダプターをプレビューまたは定義するには、クライアント属性を指定する際に `info` 属性を使用する必要があります。 `AttributeList` は、属性をコンマで区切って指定したリストです。それらの属性は、以下の順序で指定する必要があります。

`interface_type`、  
`location`、  
`interface_name`、  
`cable_type`、  
`media_speed`、  
`netaddr`、  
`subnet_mask`、  
`interface_attributes`、  
`secondary_hostname`、  
`machine_type`、  
`adapter_attributes`、  
`multiple_physloc`、  
`bos_preconfig`、  
`cust_preconfig`、  
`route`。

値が使用されないことを指定するには、小文字の `n/a` を使用します。

**-d** 2 次アダプターを定義します。有効な 2 次アダプター定義ごとに 1 個の **Client.adapter** ファイルが `adapter_def` ロケーションに作成されます。 **nimadapters** コマンドが NIM クライアントの既存の 2 次アダプター定義を検出する場合、既存の定義は置き換えられます。

**-f** `SecondaryAdapterFileName` は、2 次アダプター・ファイルの名前を指定します。

**-p** すべてのエラーを識別するためにプレビュー操作を表示します。このフラグは、2 次アダプター・ファイルまたは `info` 属性を処理しますが、NIM 環境にアダプター定義を追加しません。

プレビューでは、以下のものが表示されます。

- すべての完了した有効な 2 次アダプター・スタンザ。
- すべての無効な 2 次アダプター・スタンザおよび障害の原因。

**注:** 2 次アダプター・ファイルを使用して 2 次アダプターを構成するには、その前に **-p** フラグを使用してすべてのスタンザが正しいことを確認してください。

**-r** 2 次アダプター・スタンザ・ファイルに含まれる特定のクライアントまたはすべてのクライアントの 2 次アダプター定義を除去します。 `client` 属性または 2 次アダプター・スタンザ・ファイルを指定しない場合、`adapter_def` リソースに含まれるすべての 2 次アダプター定義が除去されます。

## パラメーター

`adapter_def` **nimadapters** コマンドを実行する場合、このパラメーターは必須です。 2 次アダプター定義ファイルが入っているディレクトリーである `adapter_def` NIM リソースを指定します。 **nimadapters** コマンドで `adapter_def` を使用するためには、その前に **nim -o define** 操作を使用することによって `adapter_def` リソースを定義しておく必要があります。

## 終了状況

- 0 コマンドは正常に実行されました。
- >0 エラーが発生しました。

## セキュリティ

root ユーザーのみがこのコマンドを実行できます。

## 例

1. 2 次アダプター定義ファイル `secondary_adapters.defs` で記述されている NIM 2 次アダプターを追加するには、次のように入力します。

```
nimadapters -d -f secondary_adapters.defs my_adapter_def
```

2. クライアント定義ファイル `secondary_adapters.defs` をプレビューするには、次のように入力します。

```
nimadapters -p -f secondary_adapters.defs my_adapter_def
```

3. `pilsner` というクライアントの NIM 2 次アダプターを定義するには、次のように入力します。

```
nimadapters -d \  
  
-a info="en,P2-I1/E1,n/a,bnc,1000_Full_Duplex,9.53.153.233,255.255.254.0,n/a,n/a,n/a,n/a,n/a" \  
  
-a client=pilsner my_adapter_def
```

4. `pilsner` というクライアントの NIM 2 次アダプター定義を `my_adapter_def` リソースから除去するには、次のように入力します。

```
nimadapters -r -a client=pilsner my_adapter_def
```

5. `secondary_adapters.defs` で定義されているクライアントの NIM 2 次アダプター定義を除去するには、次のように入力します。

```
nimadapters -r -f secondary_adapters.defs my_adapter_def
```

6. `my_adapter_def` リソースから NIM 2 次アダプター定義をすべて除去するには、次のように入力します。

```
nimadapters -r my_adapter_def
```

## ファイル

`/usr/sbin/nimadapters` `nimadapters` コマンドが入っています。

## 関連情報

`lsnim` コマンド、`nim` コマンド、`nimclient` コマンド、`nimconfig` コマンド、および `nimdef` コマンド。

インストールおよび移行の NIM マスターの構成と基本インストール・リソースの作成

---

## nimadm コマンド

### 目的

`nimadm` (Network Install Manager Alternate Disk Migration (ネットワーク・インストール・マネージャー 代替ディスク・マイグレーション)) コマンドは、システム管理者が以下のことを行うためのユーティリティです。

- rootvg のコピーを空きディスク (複数可) に作成し、同時にそれを AIX の新規バージョンまたはリリース・レベルにマイグレーションする。
- rootvg のコピーを使用して、新しいバージョンまたはリリース・レベルの AIX にマイグレーションされた新しい NIM mksysb リソースを作成する。
- NIM mksysb リソースを使用して、新しいバージョンまたはリリース・レベルの AIX にマイグレーションされた新しい NIM mksysb リソースを作成する。
- NIM mksysb リソースを使用して、空きディスク (複数の場合もある) に復元し、同時に新しいバージョンまたはリリース・レベルの AIX にマイグレーションする。

**nimadm** コマンドは、NIM リソースを使用してこれらの機能を実行します。

## 構文

代替ディスク・マイグレーションを実行する。

```
nimadm -l lpp_source -c NIMClient -s SPOT -d TargetDisks [ -a PreMigrationScript ] [ -b installp_bundle ] [ -z PostMigrationScript ] [ -e exclude_files ] [ -i image_data ] [ -j VGname ] [ -m NFSMountOptions ] [ -o bosinst_data ] [ -P Phase ] [ -j VGname ] [ -Y ] [ -F ] [ -D ] [ -E ] [ -V ] [ { -B | -r } ]
```

クライアント上の代替ディスク・マイグレーションをクリーンアップする。

```
nimadm -C -c NIMClient -s SPOT [ -F ] [ -D ] [ -E ]
```

ボリューム・グループをウェイクアップする。

```
nimadm -W -c NIMClient -s SPOT -d TargetDisks [ -m NFSMountOptions ] [ -z PostMigrationScript ] [ -F ] [ -D ] [ -E ]
```

ボリューム・グループをスリープにする。

```
nimadm -S -c NIMClient -s SPOT [ -F ] [ -D ] [ -E ]
```

代替ディスク移行ソフトウェアを同期化する。

```
nimadm -M -s SPOT -l lpp_source [ -d device ] [ -P ] [ -F ]
```

mksysb からクライアントへのマイグレーション:

```
nimadm -T NIMmksysb -c NIMClient -s SPOT -l lpp_source -d TargetDisks -j VGname -Y [ -a PreMigrationScript ] [ -b installpBundle ] [ -z PostMigrationScript ] [ -i ImageData ] [ -m NFSMountOptions ] [ -o bosinst_data ] [ -P Phase ] [ -F ] [ -D ] [ -E ] [ -V ] [ -B | -r ]
```

mksysb から mksysb へのマイグレーション:

```
nimadm -T NIMmksysb -O mksysbfile -s SPOT -l lpp_source -j VGname -Y [ -N NIMmksysb ] [ -a PreMigrationScript ] [ -b installp_bundle ] [ -z PostMigrationScript ] [ -i image_data ] [ -m NFSMountOptions ] [ -o bosinst_data ] [ -P Phase ] [ -F ] [ -D ] [ -E ] [ -V ]
```

クライアントから mksysb へのマイグレーション:

```
nimadm -c nim_client -O mksysbfile -s SPOT -l lpp_source -j VGname -Y [ -N NIMmksysb ] [ -a PreMigrationScript ] [ -b installp_bundle ] [ -z PostMigrationScript ] [ -i image_data ] [ -m NFSMountOptions ] [ -o bosinst_data ] [ -P Phase ] [ -e exclude_files ] [ -F ] [ -D ] [ -E ] [ -V ]
```

## 説明

**nimadm** コマンド (Network Install Manager Alternate Disk Migration) は、システム管理者が **rootvg** のコピーを空きディスク (複数可) に作成し、同時にそれを AIX の新規バージョンまたはリリース・レベルにマイグレーションできるようにするユーティリティです。**nimadm** コマンドは、NIM リソースを使用してこの機能を実行します。

**nimadm** コマンドを使用すると、従来型のマイグレーションに比べて、いくつかの利点があります。

1. ダウン時間が減ります。マイグレーションは、システムが立ち上がっており、正常に働いている間に行われます。インストール・メディアからブートする必要はなく、処理の大部分は NIM マスター上で行われます。
2. **nimadm** コマンドでは、マイグレーションが失敗した場合に素早くリカバリーできます。**nimadm** コマンドは **alt\_disk\_install** を使用して **rootvg** のコピーを作成するため、すべての変更はコピー (**altinst\_rootvg**) に対して行われます。重大なマイグレーション・インストール障害が起こった場合、失敗したマイグレーションはクリーンアップされ、管理者はそれ以上のアクションを取る必要はありません。AIX の新規 (マイグレーションされた) レベルで問題が起こった場合は、元のディスクからブートすることにより、システムをマイグレーション前のオペレーティング・システムに素早く戻すことができます。
3. **nimadm** コマンドを利用すると、マイグレーション・プロセスの柔軟性が高まり、カスタマイズが可能になります。これは、NIM カスタマイズ・リソース (**image\_data**、**bosinst\_data**、**exclude\_files**、**pre-migration** スクリプト、**installp\_bundle**、および **post-migration** スクリプト) を使用して行なわれます。

本書は、**nimadm** コマンドに関連する情報のみを提供していることに注意してください。

**alt\_disk\_install**、NIM、マイグレーション、およびその他のインストール関連事項の詳細については、以下の資料の最新版を参照してください。

- "インストールおよび移行"
- 「AIX Version 4.3 to 5L Migration Guide」、および IBM Redbooks® 資料

## **nimadm** ローカル・ディスク・キャッシング

ローカル・ディスク・キャッシングを使用すると、NIM マスターにおいて、クライアントへの NFS 書き込みが不要になります。これは、NFS 書き込みボトル・ネックのために **nimadm** 操作が最適に実行されない場合に便利です。 **-j VGname** フラグを指定してこの機能が呼び出された場合、**nimadm** コマンドは、指定された (NIM マスター上の) ボリューム・グループ上にファイルシステムを作成し、クライアントからそれらのすべてのデータをそれらのファイルシステムへキャッシュするためにストリームを使用します。

この機能の利点と欠点は、次のとおりです。

### 利点:

1. 比較的遅いネットワーク上での **nimadm** 操作のパフォーマンスが向上します。
2. NFS 書き込みのボトル・ネックとなっている **nimadm** 操作のパフォーマンスが向上します (NFS 書き込みには非常にコストがかかります)。
3. クライアントの CPU 使用率が下がります。
4. クライアント・ファイルシステムがエクスポートされません。

### 欠点:

1. キャッシュ・ファイルシステムは NIM マスター上でスペースを占めます (各クライアントごとに **rootvg** ファイルシステムの処理のための十分なスペースおよび移行スペースが必要)。

2. マスターの CPU 使用率が上がります。
3. マスター上の入出力が増加します (パフォーマンスをなるべく高くするには、この操作で使用される NIM リソースを含まないボリューム・グループ (ディスク) を使用してください)。

ディスク・キャッシングの実行方法:

1. NIM マスターのレベルが **bos.alt\_disk\_install.rte** 以上であることを確認してください。
2. すべての **nimadm** 操作に **-j VGName** フラグを追加してください。以下に例を示します。

```
nimadm -j rootvg ...
```

または

```
nimadm -j cachevg
```

ネットワーク上のキャッシングの対象から、特定のファイルシステム (移行に関係しない) を除外できます (それらもクライアント上の **altinst\_rootvg** にローカルにコピーされます)。ネットワーク・キャッシングから除外するファイルシステムのリストを指定するには、SPOT リソースのロケーションに、移行用のファイルを作成する必要があります。SPOT パスの正確なロケーションを調べるには、次のように入力します。

```
# lsnim -a location SpotName
```

次の形式でファイルの名前を付けてください。

```
Nim_Client.nimadm_cache.excl
```

**注:** このファイルが適用されるのは、*Nim\_Client* で指定されている NIM クライアントです。その絶対パスは、次の形式でなければなりません:

```
Spot_Location/Nim_Client.nimadm_cache.excl
```

例: **/nim\_resources/520spot/usr/myclient.nimadm\_cache.excl**

キャッシングからファイルシステムを除外するには、このファイルの中に 1 行に 1 つの (除外する) ファイルシステムを入力します。ファイルシステムを除外する時は、次の点に注意してください。

1. 移行プロセスに関係するファイルシステムは、除外しないようにしてください。つまり、それらのファイルシステムには、移行するソフトウェアが含まれています。それらを除外した場合、結果は予測不能です。
2. 次の AIX ファイルシステムは除外しないでください (除外できません)。 **/**、**/usr**、**/var**、**/opt**、**/home**、および **/tmp**

ディスク・キャッシングを使用すると、**nimadm** は次の 4 つのフェーズを変更します (それ以外のすべてのフェーズは変更されません)。

**フェーズ 2:** NIM マスターは、(NIM マスター上の) 指定されたターゲット・ボリューム・グループにローカル・キャッシュ・ファイルシステムを作成します。

**フェーズ 3:** NIM マスターは、キャッシュ・ファイルシステムにクライアントのデータを転送します。

**フェーズ 9:** NIM マスターは、移行後のすべてのデータをクライアントの代替 **rootvg** に書き込みます。

**フェーズ 10:** NIM マスターは、ローカル・キャッシュ・ファイルシステムをクリーンアップおよび除去します。



## nimadm の要件

nimadm の要件を以下に示します。

1. AIX 5.1 以上 (AIX 推奨保守レベルが 5100-03 以上) が実行されている構成済み NIM マスター。
2. NIM マスターは、**rootvg** にインストールされた **bos.alt\_disk\_install.rte** およびマイグレーションの実行に使用される SPOT と同じレベルである必要があります。(注: クライアントに **alt\_disk\_install** ユーティリティをインストールする必要はありません。)
3. 選択された **lpp\_source** NIM リソースと選択された SPOT NIM リソースでは、マイグレーション先の AIX レベルが一致していなければなりません。
4. NIM マスターの AIX のレベルは、マイグレーション先と同じか、またはそれより高いレベルでなければなりません。
5. クライアント (移行するシステム) は、AIX 4.3.3 以上でなければなりません。
6. クライアントは **rootvg** をクローンできるだけの十分なディスクを持ち、さらにマイグレーション用に (約) 500 メガの追加のフリー・スペースを持つ必要があります。必要スペースの総量は元のシステム構成と **nimadm** カスタマイズにより異なります。
7. ターゲット・クライアントは、マスターにスタンドアロン NIM クライアントとして登録しておく必要があります (詳細は **niminit** コマンドを参照)。NIM マスターは、**rshd** プロトコルを使用してクライアントにリモート・コマンドを実行できる必要があります。
8. NIM マスターは、**rshd** プロトコルを使用してクライアントにリモート・コマンドを実行できる必要があります。
9. NIM マスターとクライアントには、いずれも 128 メガバイト以上の RAM が必要です。
10. NIM マスターとクライアントの間には、大量の NFS トラフィックを処理できる信頼性の高いネットワークが必要です。NIM マスターとクライアントは、NFS マウントおよび読み取り/書き込み操作を実行できる必要があります。
11. クライアントのハードウェアとソフトウェアは、マイグレーション先の AIX レベルをサポートし、他のすべての従来のマイグレーション要件を満たす必要があります。

注: 上記の要件 1 から 10 を満たせない場合は、従来のマイグレーションを行う必要があります。要件 11 を満たせない場合は、マイグレーションはできません。

**重要:** **nimadm** マイグレーションを実行する前に、インストールされるソフトウェアについて、すべてのソフトウェア・ライセンス契約に同意する必要があります。そのためには、**nimadm** コマンドへの引数として **-Y** フラグを指定するか、または **ADM\_ACCEPT\_LICENSES** 環境変数を「yes」に設定します。

## nimadm の制限

**nimadm** コマンドには以下の制限があります。

1. クライアントの **rootvg** で TCB がオンになっている場合は、それを (永続的に) 使用不可にするか、ディスク・キャッシュ・オプション (**-j**) を使用するか、従来型のマイグレーションを実行する必要があります。(この制限がある理由は、TCB は、NFS に見えないファイル・メタデータをアクセスする必要があるからです。)
2. **nimadm** コマンドが使用するすべての NIM リソースは、NIM マスターから見てローカルである必要があります。
3. マイグレーション中に、クライアントのアクティブな **rootvg** に干渉することはほとんどありませんが、ディスク入出力、biod 活動、および **alt\_disk\_install** クローン作成にともなう CPU 使用の増加により、若干のパフォーマンス低下が発生する可能性があります。
4. **nimadm** パフォーマンスを最適化するため、NFS チューニングが必要になる場合があります。

5. **nimadm** コマンドは、**bos\_hd5** 論理ボリュームがある場合、**multibos** コマンドと一緒にサポートされません。

## **nimadm** が使用する NIM リソース

### SPOT リソース (-s フラグ)

NIM スポット・リソースは、すべての **nimadm** 操作 (マイグレーション、クリーンアップ、ウェイクアップ、スリープ) に必要です。クライアントが使用するすべての **nimadm** および **alt\_disk\_install** ユーティリティーは、このリソースにインストールされます。クライアントに **nimadm** ソフトウェアをインストールする必要はありません。以下のファイルセットをスポットにインストールするには、NIM の **cust** 操作を使用する必要があります。

- 必須: **bos.alt\_disk\_install.rte** (NIM マスターのレベルと一致する必要があります)
- オプションのメッセージ・カタログ: **bos.msg.\$LANG.alt\_disk\_install.rte**

### lpp\_source リソース (-l フラグ)

この NIM リソースは、システムのマイグレーションに使用されるインストール・イメージのソースです。これは **nimadm** マイグレーション操作に必要です。 **lpp\_source** は、マイグレーションされるレベルのすべてのシステム・イメージを含む必要があります (**lsnim -l lpp\_source** 出力の **lpp\_source images** 属性を検査してください)。またここには、マイグレーションに必要なすべてのオプションの **installp** イメージも含める必要があります。

### pre-migration

このスクリプト・リソースは NIM マスターで実行されますが、マスターにマウントされる、クライアントの **alt\_inst** ファイルシステムの環境内で実行されます (これは **chroot** コマンドを使用して実行されます)。このスクリプトはマイグレーションを開始する前に実行されます。

### post-migration

このスクリプトは **pre-migration** スクリプトに似ていますが、マイグレーションが完了した後に実行されます。

### image\_data

**alt\_disk\_install** に渡される **image\_data** リソースを指定します (-i フラグへの引数として渡される)。NIM は **alt\_disk\_install** を呼び出す前に、クライアント上でこのリソースを割り当て、マウントします。

### exclude\_files

**alt\_disk\_install** に渡される **exclude\_files** リソースを指定します (-e フラグの引数として渡される)。NIM は **alt\_disk\_install** を呼び出す前に、クライアント上でこのリソースを割り当て、マウントします。

### installp\_bundle

この NIM リソースは、マイグレーションが完了した後で **nimadm** コマンドがインストールする追加ソフトウェアを指定します。

### bosinst\_data

この NIM リソースは、**nimadm** コマンドが使用する各種のインストール設定値を指定します。

## **nimadm** のマイグレーション・プロセス

**nimadm** コマンドはマイグレーションを 12 のフェーズで実行します。各フェーズは、**-P** フラグを使用して個別に実行することができます。マイグレーションをフェーズ別に実行する前に、**nimadm** プロセスについてよく理解しておく必要があります。**nimadm** フェーズを以下に示します。

1. マスターはクライアントに **alt\_disk\_install** コマンドを実行し、これにより **rootvg** のコピーがターゲット・ディスクに作成されます (同時に、**alt\_disk\_install** プロセスのフェーズ 1 です)。このフェ

ーズで、**altinst\_rootvg** (代替 rootvg) が作成されます。ターゲット **mksysb** が指定されている場合は、NIM マスター上のローカル・ディスク・キャッシングを使用して、**mksysb** を用いて **rootvg** が作成されます。

2. マスターはリモート・クライアント・コマンドを実行し、すべての **/alt\_inst** ファイルシステムをマスターにエクスポートします。ファイルシステムはマスターへのルート・アクセスを持つ読み取り/書き込みとしてエクスポートされます。ターゲット **mksysb** が指定されている場合、キャッシュ・ファイルシステムが、**mksysb** のイメージ・データを基にして、作成されます。
3. マスター NFS は、フェーズ 2 でエクスポートされたファイルシステムをマウントします。ターゲット **mksysb** が指定されている場合は、**mksysb** アーカイブが、フェーズ 2 で作成されたキャッシュ・ファイルシステム内に復元されます。
4. **pre-migration** スクリプト・リソースを指定した場合は、この時点で実行されます。
5. システム構成ファイルが保管されます。初期マイグレーション・スペースを計算し、これに相当するファイルシステム拡張が行われます。「bos」を復元し、装置データベースをマージします (従来のマイグレーションと同様)。マイグレーション・マージ方式のすべてが実行され、いくつかの各種処理が行われます。
6. すべてのシステム・ファイルセットが **installp** を使用してマイグレーションされます。このフェーズ中に、必要な RPM イメージもインストールされます。
7. **post-migration** スクリプト・リソースを指定した場合は、この時点で実行されます。
8. **bosboot** を実行してクライアントのブート・イメージを作成し、これをクライアントのブート論理ボリューム (**hd5**) に書き出します。
9. フェーズ 3 でマスターに行われたマウントをすべて取り外します。
10. フェーズ 2 で作成されたすべてのクライアント・エクスポートを除去します。
11. **alt\_disk\_install** を再度呼び出し (**alt\_disk\_install** のフェーズ 3)、最終調整を行い、**altinst\_rootvg** をスリープします。 **bootlist** がターゲット・ディスクに設定されます (**-B** フラグが指定されていない場合)。出力 **mksysb** が指定されている場合は、キャッシュが、**mksysb** ファイルにアーカイブされ、そして NIM **mksysb** リソースとなります。
12. クリーンアップを実行し、マイグレーションを終了します。 **-r** フラグを指定した場合、クライアントはリポートされます。

注: **nimadm** コマンドは複数クライアントの同時マイグレーションをサポートします。

## nimadm クリーンアップ操作

この操作は「**-C**」フラグで示され、何らかの理由でマイグレーションが失敗し、クリーンアップが実行されなかった場合に、クリーンアップを実行するように設計されています。また新しいマイグレーションを実行するために、前のマイグレーションをクリアするためにも使用できます。

## nimadm ウェイクアップおよびスリープ

マイグレーションが完了した後、**nimadm** コマンドを使用して、マイグレーションされた **altinst\_rootvg** または元の **rootvg** (マイグレーションされたディスクからブートされた場合) を「ウェイクアップ」することができます。 **nimadm** のウェイクアップ (**-W** フラグ) は **alt\_disk\_install** ウェイクアップを実行し、NFS は **/alt\_inst** ファイルシステムをエクスポートして、これを NIM マスターにマウントします。 **nimadm** のスリープ機能 (**-S** フラグ) はウェイクアップを逆に行うものであり、これを行うには、NIM マスターのマウントをアンマウントし、 **/alt\_inst** ファイルシステムをアンエクスポートし、クライアントに **alt\_disk\_install** のスリープ機能を実行します。

## フラグ

<b>-a</b> <i>PreMigrationScript</i>	pre-migration NIM スクリプト・リソースを指定します。
<b>-b</b> <i>installp_bundle</i>	installp_bundle NIM リソースを指定します。
<b>-B</b>	<b>nimadm</b> マイグレーションの後に <b>bootlist</b> を実行しないことを指定します。これを設定した場合は、 <b>-r</b> フラグは使用できません。
<b>-c</b> <i>TargetDisks</i>	この <b>nimadm</b> 操作のターゲットとなる NIM 定義クライアントを指定します。このフラグはすべての <b>nimadm</b> 操作に必要です。
<b>-C</b>	<b>nimadm</b> クリーンアップを実行します。
<b>-d</b> <i>TargetDisks</i>	<b>altinst_rootvg</b> (マイグレーションされるボリューム・グループ) の作成に使用するクライアント・ターゲット・ディスクを指定します。
<b>-D</b>	<b>nimadm</b> コマンドをデバッグ・モードに設定します。この機能は <b>nimadm</b> 関連の問題をデバッグする際に使用し、デフォルトに設定すべきではありません。
<b>-e</b> <i>exclude_files</i>	<b>exclude_files</b> NIM リソースを指定します。このリソースは <b>alt_disk_install</b> コマンドでフェーズ 1 中に使用されます。
<b>-E</b>	重大なマイグレーション・エラーが起こった場合、 <b>nimadm</b> デバッガーに入ります。
<b>-F</b>	クライアントを強制的にアンロックします。通常、 <b>nimadm</b> コマンドは、各種の操作を実行するためにクライアントをロックします。クライアントがロックされている間、他の <b>nimadm</b> または NIM 操作は実行できません。このフラグは、クライアントが誤ってロックされた場合 (例えば、なんらかの理由で <b>nimadm</b> コマンドが障害の後でクリーンアップを呼び出せなかった場合) などの、例外的な状況でのみ使用してください。
<b>-i</b> <i>image_data</i>	<b>image_data</b> NIM リソースを指定します。このリソースは <b>alt_disk_install</b> コマンドでフェーズ 1 およびフェーズ 11 中に使用されます。
<b>-j</b> <i>VGname</i>	指定された (NIM マスター上の) ボリューム・グループ上にファイルシステムを作成し、クライアントからそれらのファイルシステムへのすべてのデータのキャッシングのためにストリームを使用します。
<b>-l</b> <i>lpp_source</i>	この <b>nimadm</b> 操作で使用される <b>lpp_source</b> NIM リソースを指定します。このフラグはマイグレーション操作には必須です。
<b>-m</b> <i>NFSMountOptions</i>	クライアントのリソースをマスター上にマウントする、 <b>mount</b> コマンドに渡す引数を指定します。このフラグは、 <b>nimadm</b> に関係する NFS パフォーマンスをチューニングするために使用できます。
<b>-M</b>	NIM マスター、SPOT、 <b>lpp_source</b> 、およびオプション・デバイス上の <b>alt_disk_install</b> ソフトウェア ( <b>bos.alt_disk_install</b> ) の各レベルが同期 (一致) していることを検査します。一致しない場合、 <b>nimadm</b> コマンドは、 <b>lpp_source</b> またはオプション・デバイスで検出された最高レベルのソフトウェアをインストールします。
<b>-N</b> <i>NIMmksysb</i>	作成する固有の新規 NIM <b>mksysb</b> リソースを指定します。 <b>-N</b> フラグを指定した場合は、 <b>-O</b> フラグも指定する必要があります。
<b>-o</b> <i>bosinst_data</i>	<b>bosinst_data</b> NIM リソースを指定します。
<b>-O</b> <i>mksysbfile</i>	マイグレーション済み <b>mksysb</b> のファイル・パス名を指定します。 <b>-O</b> フラグを指定した場合は、 <b>-j</b> フラグ、および <b>-c</b> か <b>-T</b> のどちらかのフラグを指定する必要があります。
<b>-P</b> <i>Phase</i>	<b>nimadm</b> コマンドのこの呼び出し中に実行するフェーズ。複数のフェーズを実行する場合は、フェーズをスペースまたはコンマで区切って指定する必要があります。有効なフェーズは 1 から 12 です。
<b>-r</b>	<b>nimadm</b> マイグレーションが完了した後、クライアントがリポートすべきことを指定します。
<b>-s</b> <i>SPOT</i>	この <b>nimadm</b> 操作に使用される <b>SPOT</b> NIM リソースを指定します。このフラグはすべての <b>nimadm</b> 操作に必要です。
<b>-S</b>	<b>nimadm</b> の「スリープ」機能を実行します。この機能は、 <b>nimadm</b> の「ウェイクアップ」を終了するために実行する必要があります。

<b>-T</b> <i>NIMmksysb</i>	マイグレーションする既存の NIM mksysb リソースを指定します。 <b>-T</b> フラグを指定した場合は、 <b>-j</b> フラグ、および <b>-O</b> か <b>-c</b> のどちらかのフラグを指定する必要があります。
<b>-V</b>	詳細出力をオンにします。
<b>-W</b>	<b>nimadm</b> の「ウェイクアップ」機能を実行します。
<b>-Y</b>	インストールされるソフトウェアの必要なソフトウェア・ライセンス契約に同意します。
<b>-z</b> <i>PostMigrationScript</i>	post-migration NIM スクリプト・リソースを指定します。

## 終了状況

- 0 すべての **nimadm** コマンド関連操作が正常に完了しました。
- >0 エラーが発生しました。

## セキュリティ

root ユーザーだけが **nimadm** コマンドを実行できます。

## 例

- NIM **SPOT** リソース spot1、NIM **lpp\_source** リソース lpp1、およびターゲット・ディスク hdisk1 と hdisk2 を使用して、ターゲット NIM クライアント aix1 に **nimadm** マイグレーションを実行します。 **-Y** フラグは、インストールするソフトウェアのすべての必要なソフトウェア・ライセンス契約に同意することに注意してください。次のように入力します。

```
nimadm -c aix1 -s spot1 -l lpp1 -d "hdisk1 hdisk2" -Y
```

- 上の例と同じ操作を hdisk2 に実行し、pre-migration スクリプト nimscript1 と post-migration スクリプト nimscript2 を実行するには、次のように入力します。

```
nimadm -c aix1 -s spot1 -a nimscript1 -z nimscript2 -l lpp1 -d hdisk1 -Y
```

- NIM **SPOT** リソース spot1 を使用して、クライアント aix1 に **nimadm** クリーンアップを実行するには、次のように入力します。

```
nimadm -C -c aix1 -s spot1
```

- ファイル名 nim1 を使用してクライアントのマイグレーション済み新規 mksysb リソースを作成するには、次のように入力します。

```
nimadm -c aix1 -s spot1 -l lpp1 -O /export/mksysb/mksysb1 -j vg00 -Y -N nim1
```

- 既存の NIM mksysb リソースからファイル名 nim3 を使用してマイグレーション済み新規 mksysb リソースを作成するには、次のように入力します。

```
nimadm -s spot1 -l lpp1 -j vg00 -Y -T nim2 -O /export/mksysb/m2 -N nim3
```

- 既存の NIM リソースをマイグレーションし、これをクライアント上に置くには、次のように入力します。

```
nimadm -c aix1 -s spot1 -l lpp1 -d hdisk1 -j vg00 -T nim2 -Y
```

注: nim2 NIM mksysb リソースには変更が加えられません。

## ファイル

**/usr/sbin/nimadm** **nimadm** コマンドが入っています。



## 関連情報

lspp コマンド、nim コマンド、lsnim コマンド、alt\_disk\_install コマンド、installp コマンド、chroot コマンド。

---

## nimclient コマンド

### 目的

ネットワーク・インストール管理 (NIM) 操作を NIM クライアントから実行できるようにします。

### 構文

#### NIM マスターのプッシュ許可を使用可能または使用不可にする

```
nimclient { -p } | { -P }
```

#### NIM マスター・プッシュ操作の暗号認証を使用可能または使用不可にする

```
nimclient { -c } | { -C }
```

#### NIM 環境の情報をリストする

```
nimclient -l LsnimParameters
```

#### 日時を NIM マスターの日時に設定する

```
nimclient -d
```

#### NIM 操作を行う

```
nimclient -o Operation [ -a Attribute=Value] ...
```

### 説明

**nimclient** コマンドは、NIM クライアントのワークステーションで NIM リソースを取り出すために使用します。このコマンドを使用すると、NIM マスター・サーバーでワークステーションをインストールおよびカスタマイズする機能を使用可能にしたり使用不可にしたりできます。**nimclient** コマンドは、使用可能な NIM リソースのリストを生成したり、既にクライアントに割り当てられている NIM リソースを表示したりするために使用します。**-o** フラグを指定して **nimclient** コマンドを実行すると、いくつかの NIM 操作を行えますが、行える操作は限られています。

### フラグ

**-a Attribute=Value**

NIM 操作に情報を渡します。

マスターから

**lsnim -q Operation -t Type** コマンドを使用して、特定の操作に有効な属性のリストを入手します。

クライアントから

**nimclient -l -q Operation -t Type** コマンドを使用して、特定の操作に有効な属性のリストを入手します。

- c** NIM マスター・プッシュ操作の際に SSL 認証を使用可能にします。  
**注:** **nimconfig -c** コマンドを使用して、NIM マスター上に OpenSSL 証明書  
構成する必要があります。SSL 証明書は、**nimclient -c** が実行されるとき  
に、NIM マスターからコピーされます。
- C** SSL 認証を使用不可にし、NIM マスター・プッシュ操作の際に標準の **nimsh**  
セキュリティを使用します。
- d** クライアントの日時をマスターの日時に設定します。
- l Lsnim parameters** 指定した **lsnim** パラメーターを使用して、マスターで **lsnim** コマンドを実行  
します。このオプションを指定して使用するパラメーターはすべて、**lsnim** コ  
マンドの構文規則に従います。**lsnim** 構文の中には、NIM オブジェクト名を  
使用しなければならないものもあります。ご使用のマシンの NIM 名は、  
**/etc/niminfo** ファイルで調べてください。
- o Operation** 指定された操作を行います。実行できる操作は次のとおりです。
- allocate**  
リソースを割り当てて使用できるようにします。
- bos\_inst**  
BOS インストールを実行します。
- change**  
オブジェクトの属性を変更します。
- check** NIM オブジェクトの状況を検査します。
- cust** ソフトウェアのカスタマイズを実行します。
- deallocate**  
リソースの割り当てを解除します。
- diag** マシンが診断イメージをブートできるようにします。
- maint\_boot**  
マシンが保守モードでブートできるようにします。
- reset** オブジェクトの NIM 状態をリセットします。
- showres**  
NIM リソースの内容を表示します。
- p** NIM マスターがコマンドをプッシュできるようにします。
- P** コマンドをプッシュするための NIM マスターの権限を除去します。  
**注:** マスターは、**-F** フラグを使用してこの制限を変更できます。

## セキュリティ

アクセス制御: **nimclient** コマンドを実行するには、**root** 権限が必要です。

## 例

1. マシンの NIM 名が **pluto** である場合にこのマシンで使用できるすべての NIM リソースをリストする  
には、次のように入力します。

```
nimclient -l -L pluto
```

2. マシンの NIM 名が **pluto** である場合にこのマシンで使用できるすべての共用プロダクト・オブジェ  
クト・ツリー (SPOT) をリストするには、次のように入力します。

```
nimclient -l -L -t spot pluto
```

3. このマシンから開始できる操作をリストするには、次のように入力します。

```
nimclient -l -p -s pull_ops
```



4. NIM マスターがクライアントでローカルにコマンドを実行しないようにするには、次のように入力します。

```
nimclient -P
```

5. `myspot` という名前のスポット・リソース、`images` という名前の **lpp\_source** リソース、および `dept_bundle` という名前の **installp** バンドル・ファイルを割り当てるには、次のように入力します。

```
nimclient -o allocate -a spot=myspot -a lpp_source=images \  
-a installp_bundle=dept_bundle
```

6. 必要なリソースを割り当てたあとで基本システム・インストールを実行するには、次のように入力します。

```
nimclient -o bos_inst
```

7. スタンドアロン・クライアントから、**lpp\_source** を割り当てて、インストール可能なオプション `adt` のイメージが **lpp\_source** の `images` にあるようなソフトウェア・プロダクトをインストールするには、次のように入力します。

```
nimclient -o allocate -a lpp_source=images
```

それから、次のように入力します。

```
nimclient -o cust -a filesets="adt"
```

8. スタンドアロン・クライアントから、**lpp\_source** を割り当てて、インストール可能なオプション `adt` のイメージが **lpp\_source** の `images` にあり、インストール可能なオプションの名前が **installp\_bundle** の `bundle3` にあるようなソフトウェア・プロダクトをインストールするには、次のように入力します。

```
nimclient -o allocate -a lpp_source=images \  
-a installp_bundle=bundle3
```

それから、次のように入力します。

```
nimclient -o cust
```

9. APAR `IX12345` に関連するすべてのファイルセット更新をインストールし、**lpp\_sourceupdt\_images** に常駐させるには、次のように入力します。

```
nimclient -o allocate -a lpp_source=updt_images  
nimclient -o cust -afixes=IX12345
```

10. クライアントにインストールされているすべてのソフトウェアを `updt_imageslpp_source` から最新の更新を使って更新するには、次のように入力します。

```
nimclient -o allocate -a lpp_source=updt_images  
nimclient -o cust -afixes=update_all
```

11. `spot1` という名前の **SPOT** リソースを使って、システムが保守モードでブートできるようにするには、次のように入力します。

```
nimclient -o maint_boot -a spot=spot1
```

これによって保守ブート操作が設定されますが、ネットワーク・ブートはローカルで開始する必要があります。

12. `config` スクリプト `script1` の内容を表示するには、次のように入力します。

```
nimclient -o showres -a resource=script1
```

13. `bosinst.data` リソース `bosinst_data1` の内容を表示するには、次のように入力します。

```
nimclient -o showres -a resource=bosinst_data1
```

14. NIM クライアント・マシン `machine1` からマシン `machine1` に現在インストールされているものに関連する `lpp_source lpp_source1` 内のファイルセットをすべてリストするには、次のように入力します。

```
nimclient -o showres -a resource=lpp_source1
```

**reference** 属性は、**nimclient** コマンドによって自動的に提供されます。

15. `lpp_source lpp_source1` 上の `bos.INed` ファイルセットおよび `x1C.rte` ファイルセットに対するユーザー命令をリストするには、次のように入力します。

```
nimclient -o showres -a filesets="bos.INed x1C.rte" \  
-a resource=lpp_source1 -a installp_flags="qi"
```

16. `lpp_source1` という `lpp_source` 上のソフトウェアによって修正されたすべての問題をリストするには、次のように入力します。

```
nimclient -o showres -a instfix_flags="T" -a resource=lpp_source1
```

17. インストール操作でリソースを割り当てている間に、**lpp\_sourceclient\_images** を使用して NIM **installp\_bundleclient\_bundle** にリストされているファイルセットをインストールするには、次のように入力します。

```
nimclient -o cust -a installp_bundle=client_bundle \  
-a lpp_source=client_images
```

18. `client_grp` という名前の NIM リソース・グループから利用可能なすべてのリソースを自動的に割り当てている間に、基本システムのインストールを実行するには、次のように入力します。

```
nimclient -o bos_inst -a group=client_grp
```

19. マスターでデフォルト・リソース・グループとして定義されている NIM グループから利用可能なすべてのリソースを自動的に割り当てている間に、基本システムのインストールを実行するには、次のように入力します。

```
nimclient -o bos_inst
```

20. SSL 証明書をコピーして SSL 認証を使用可能にするには、次のように入力します。

```
nimclient -c
```

注: このコマンド・オプションを使用する前に、NIM クライアント上に OpenSSL がインストールされている必要があります。

## ファイル

`/etc/niminfo` NIM が使用する変数が入っています。

## 関連情報

`lsnim` コマンド、`nim` コマンド、`nimconfig` コマンド、`niminit` コマンド。

`.info` ファイル。

---

## nimconfig コマンド

### 目的

ネットワーク・インストール管理 (NIM) マスター・パッケージを初期化します。

## 構文

### NIM マスター・パッケージを初期化する

```
nimconfig -a pif_name=Pif -a netname=Objectname [ -a master_port=PortNumber ] [ -a platform=Value ] [ -a registration_port=PortNumber ] [-a ring_speed=Speed | -a cable_type=CableType ]
```

### NIM 環境の SSL を構成する

```
nimconfig -c
```

### /etc/niminfo ファイルを再作成する

```
nimconfig -r
```

## 説明

**nimconfig** コマンドは、NIM マスター・パッケージを初期化します。このパッケージを初期化しないと、他のいずれの NIM コマンドも使用できません。 **-a** フラグを使用して適切な属性を指定すると、

**nimconfig** コマンドは次のタスクを実行して NIM 環境を初期化します。

- *ObjectName* パラメーターで指定されたネットワーク・オブジェクトを定義して、 *Pif* パラメーターで指定された NIM マスターの 1 次インターフェースを接続するネットワークを表します。
- 新しく定義したネットワーク・オブジェクトに接続して、NIM マスターの定義を完了します。
- ネットワーク・ブート・リソースを表すリソース・オブジェクトを定義します。このネットワーク・ブート・リソースは、NIM によって自動的に管理されます。
- カスタマイズを行うために NIM によって自動的に作成されるカスタマイズ・スクリプトを表す、リソース・オブジェクトを定義します。
- NIM コミュニケーション・デーモン、 **nimesis** を開始します。

## フラグ

**-a** 次の attribute=value の組を割り当てます。

**pif\_name=***Pif*

NIM マスターの 1 次ネットワーク・インターフェースを指定します。この値は、使用可能状態にある論理インターフェース名 (tr0 や en0 など) でなければなりません。

**master\_port=***PortNumber*

NIM クライアント通信に使用する **nimesis** デーモンのポート番号を指定します。

**platform=***Value*

プラットフォームを指定します。サポートされているプラットフォームは次のとおりです。

**rs6K** AIX 5.1 およびそれ以前用のマイクロチャネル・ベースのユニプロセッサ・モデル

**rs6ksmp**

AIX 5.1 およびそれ以前用のマイクロチャネル・ベースの対称型マルチプロセッサ・モデル

**rspc** AIX 5.1 およびそれ以前用の PowerPC PCI バス・ベースのユニプロセッサ・モデル

**rspsmp**

AIX 5.1 およびそれ以前用の PowerPC PCI バス・ベースの対称型マルチプロセッサ・モデル

**netname=***ObjectName*

マスターの 1 次インターフェースを接続するネットワークを表すネットワーク・オブジェクトの作成時に、**nimconfig** コマンドで使用する名前を指定します。

**ring\_speed=***Speed*

Mbps 単位のスピード。**pif\_name** がトークンリング・ネットワークを参照する場合は、この値が必要です。指定できる値は次のとおりです。

4

16

**cable\_type=***CableType*

イーサネット・ケーブル・タイプを指定します。**pif\_name** がイーサネット・ネットワークを参照する場合は、この値が必要です。指定できる値は次のとおりです。

bnc

dix

N/A

**registration\_port=***PortNumber*

NIM クライアント登録に使用するポート番号を指定します。

**注:** コマンド行でポート番号を指定しない場合は、NIM の **/etc/services** ファイルのポート番号が使用されます。**/etc/services** ファイルに NIM ポート **nim** および **nimreg** のエントリがない場合、**master\_port** では 1058、**registration\_port** では 1059 のデフォルト値が使用されます。

**-c** OpenSSL が NIM マスター上にインストール済みの場合は、NIM クライアント通信の際に使用する SSL キーと証明書が、このオプションで作成されます。SSL 証明書は後で、**nimclient -c** コマンドを使用して NIM クライアントにコピーされます。

**-r** NIM データベースに既に存在する情報を使用して、マスター上に **/etc/niminfo** ファイルを再作成します。このマシンで **bos.sysmgmt.nim.master** パッケージが構成されていない場合、このオプションは失敗します。このオプションは、**/etc/niminfo** ファイルを間違えて除去してしまった場合に使用してください。

## セキュリティ

アクセス制御: **nimconfig** コマンドを実行するには、**root** 権限が必要です。

### 例

1. トークンリングおよびネットワーク・コミュニケーション用のデフォルト NIM ポートを使用して NIM 環境を初期化するには、次のように入力します。

```
nimconfig -a pif_name=tr0 -a netname=net1 -a ring_speed=16
```

2. イーサネットおよびデフォルト NIM ポートを使用して NIM 環境を初期化するには、次のように入力します。

```
nimconfig -a pif_name=en0 -a master_port=1058 \  
-a netname = net2 -a cable_type=bnc
```

3. マシンが既にマスターとして正しく構成されている場合に、その NIM マスター上で **/etc/niminfo** ファイルを再作成するには、次のように入力します。

```
nimconfig -r
```

4. ATM ネットワーク・インターフェースを使用して NIM マスターを初期化するには、次のように入力します。

```
nimconfig -a pif_name=at0 -a master_port=1058 -a netname=ATMnet
```

**注:** ATM ネットワークへのインターフェースは、現在ネットワーク上のブートをサポートしていないので、この操作では、マスターのサブネットに対応する汎用ネットワーク・オブジェクトを定義します。

5. NIM クライアント・コミュニケーションの TCP/IP ポート 1060 および NIM クライアント登録の TCP/IP ポート 1061 を使用して NIM 環境を初期化するには、次のように入力します。

```
nimconfig -a pif_name=tr0 -a netname=net2 -a master_port=1060 \  
-a registration_port=1061 -a ring_speed=16
```

6. NIM 通信用の SSL キーおよび証明書を作成するには、次のように入力します。

```
nimconfig -c
```

**注:** このコマンド・オプションを使用する前に、NIM マスター上に OpenSSL がインストールされている必要があります。

## ファイル

**/etc/niminfo** NIM が使用する変数が入っています。

## 関連情報

**lsnim** コマンド、**nim** コマンド、**nimclient** コマンド、**niminit** コマンド。

**.info** ファイル。

---

## nimdef コマンド

### 目的

ネットワーク・インストール管理 (NIM) クライアントをスタンザ・ファイルから定義します。

## 構文

**nimdef** [ **-p** | **-d** | **-c** ] **-f** *Name*

## 説明

**nimdef** コマンドは、定義スタンザ・ファイルを構文解析して、NIM 環境に NIM クライアント定義を追加するために必要なコマンドを作成します。

また、**nimdef** コマンドは、新しいクライアント定義をサポートするために NIM 環境で自動的に NIM ネットワークおよび NIM マシン・グループを作成します。

**注:** **nimdef** コマンドを使用する前に、NIM マスターを構成する必要があります。(詳細については、インストールおよび移行の基本 NIM 操作および構成を参照してください。)

## クライアント定義ファイルの規則

クライアント定義ファイルのフォーマットには、次の規則があります。

- スタンザ・ヘッダーのあとに、*Attribute = Value* 形式の属性行を続けます。
- 同じスタンザ内で、ある属性値を複数回定義すると、属性が **machine\_group** である場合を除き、最後の定義だけが使用されます。**machine\_group** 属性を複数定義すると、そのすべてがマシン定義に適用されます。
- 無効な属性キーワードを使用すると、その属性定義は無視されます。
- ファイルの各行に入れられるのは、1 つのヘッダーまたは属性定義だけです。
- 各マシン・ホスト名の定義ファイルに入れられるスタンザは 1 つだけです。
- スタンザ・ヘッダー・エントリでキーワード **default** を指定すると、デフォルト値を定義できます。
- マシン・ホスト名を除くすべてのマシン属性に対してデフォルト値を指定できます。マシンの属性を指定せず、デフォルト値を定義すると、そのデフォルト値が使用されます。
- デフォルト値は、定義ファイル内のどのロケーションでも指定または変更できます。デフォルト値は、設定が済むと、それ以降のすべての定義に適用されます。
- あとに続くすべてのマシン定義に対するデフォルト値をオフにするには、デフォルト・スタンザで属性値を **nothing** に設定します。
- 単一のマシン定義に対するデフォルト値をオフにするには、マシン・スタンザで属性値を **nothing** に設定します。
- クライアント定義ファイルにはコメントを入れることができます。コメントは、ポンド (#) 文字で開始します。
- ヘッダーや属性のキーワードおよび値の定義ファイルを構文解析する際、タブ文字およびスペースは無視されます。

## クライアント定義ファイルのキーワード

クライアント定義ファイルでは、次のキーワードを使用してマシン属性を指定します。

### 必須属性

<b>cable_type</b>	マシンのケーブル・タイプを指定します。 <b>network_type</b> が <b>ent</b> である場合に必須です。
<b>gateway</b>	マシンが使用するデフォルト・ゲートウェイのホスト名または IP アドレスを指定します。ゲートウェイを使用しないマシンの場合は、この属性に値 <b>0</b> (ゼロ) を指定します。
<b>machine_type</b>	マシンのタイプを指定します。 <b>standalone</b> 、 <b>diskless</b> 、または <b>dataless</b> です。
<b>network_type</b>	マシンのネットワーク・アダプターのタイプを指定します。 <b>ent</b> または <b>tok</b> です。

**ring\_speed**                    マシンのリング・スピードを指定します。 **network\_type** が **tok** である場合に必須です。  
**subnet\_mask**                マシンが使用するサブネット・マスクを指定します。

## オプションの属性

<b>nim_name</b>	マシンに使用する NIM 名を指定します。この属性は、ホスト名以外の名前を NIM 名として使用する場合に指定します。デフォルトでは、マシンに対して指定した NIM 名からすべてのドメイン情報を取り除いたものが、マシンのホスト名になります。別々のドメインで非固有のホスト名を使用した場合、どちらのマシンでも同じ NIM 名が使用されているため、競合が起きます。そのような環境では、影響を受けるマシン定義に対してこの属性を定義してください。
<b>platform</b>	マシン・ハードウェア・プラットフォームを指定します。この属性を指定しない場合、AIX 5.1 までのみのデフォルトは <b>rs6k</b> です。
<b>net_adptr_name</b>	マシンが使用するネットワーク・アダプターの名前を指定します ( <b>tok0</b> 、 <b>ent0</b> 、など)。
<b>netboot_kernel=NetbootKernelType</b>	ネットワーク上でクライアントをブートする際に使用するカーネルのタイプを指定します。 <b>netboot_kernel</b> 値は <b>up</b> または <b>mp</b> です。
<b>ipl_rom_emulation</b>	IPL ROM エミュレーションに使用するデバイスを指定します ( <b>/dev/fd0</b> 、 <b>/dev/rmt0</b> など)。
<b>primary_interface</b>	オリジナルのマシン定義に使用するホスト名を指定します。この属性は、現在のスタンザで、NIM 環境で定義されているマシンへの追加のインターフェースだけを定義する場合に使用します。
<b>master_gateway</b>	別のネットワーク上にあるそのマシンにアクセスするために NIM マスターが使用するゲートウェイを指定します。マシンが NIM 環境で既に定義されているネットワーク上で定義されている場合や、NIM マスター・ネットワークにデフォルト・ゲートウェイが指定されている場合、この属性は必要ありません。
<b>machine_group</b> <b>comments</b>	定義されているマシンを追加するグループを 1 つまたは複数指定します。マシン定義に入れるコメントを指定します。コメント文字列は二重引用符 (") で囲みます。

## クライアント定義ファイルのスタンザ・エラー

次の条件下では、定義スタンザは誤りです。

- 定義のスタンザ・ヘッダーで使用されているホスト名が解決不能。
- 必須の属性が存在しない。
- 属性に無効値を指定している。
- 属性ミスマッチが起こっている。例えば、同じスタンザで **network\_type=tok** と **cable\_type=bnc** を指定することはできません。
- グループ・タイプのミスマッチが起こっている。例えば、グループにスタンドアロン・マシンが含まれていて、**machine\_type=diskless** を指定している場合、マシンに対してグループを指定することはできません。
- 同じホスト名に対してマシン定義を複数回行っている。
- NIM 環境で既に定義されているマシンに対してマシン定義を行っている。
- マシン定義の **primary\_interface** 値が、定義されたマシンまたはスタンザ定義のホスト名と一致しない。
- マシン定義の **primary\_interface** 値が別のマシン定義のホスト名と一致しているが、その定義が誤っている。



## サンプル・クライアント定義ファイル

以下のデフォルト値は、AIX 5.1 およびそれ以前の場合です。

```
# Set default values.
default:
    machine_type = standalone
    subnet_mask  = 255.255.240.0
    gateway      = gateway1
    network_type = tok
    ring_speed   = 16
    platform     = rs6k
    machine_group = all_machines

# Define the machine "lab1"
# Take all defaults.
lab1:

# Define the machine "lab2"
# Take all defaults and specify 2 additional attributes.
# The machine "lab2" uses IPL ROM emulation, and will be added to
# the machine groups "all_machines" and "lab_machines".
lab2:
    ipl_rom_emulation = /dev/fd0
    machine_group      = lab_machines

# Define the machine "lab3"
# Take all defaults, but do not add the machine to the default
# group.
lab3:
    machine_group=

# Define the machine "lab4"
# Take all defaults, but do not add "lab4" to the default group
# "all_machines".
# Instead add it to the groups "lab_machines" and "new_machines".
lab4:
    machine_group =
    machine_group = lab_machines
    machine_group = new_machines

# Change the default "platform" attribute.
default:
    platform = rspc

# define the machine "test1"
# Take all defaults and include a comment.
test1:
    comments = "This machine is a test machine."
```

## フラグ

- c** クライアント定義ファイルからコマンドを生成します。このフラグは、定義ファイルを処理し、定義を追加するためのコマンドを生成します。コマンドは呼び出されませんが、KSH スクリプトとして表示されるので、ファイルにリダイレクトして、あとで呼び出すことができます。
- d** クライアント定義ファイルからマシンを定義します。このフラグは、定義ファイルを処理し、定義を追加するためのコマンドを NIM 環境に呼び出します。
- f Name** クライアント定義ファイルの名前を指定します。

- p** クライアント定義ファイルのプレビューを表示します。このフラグは、定義ファイルを処理しますが、NIM 環境にマシンを追加しません。次のように表示されます。
- すべての完了した有効な NIM 定義スタンザ。
  - マシン用に定義するすべての追加インターフェース。
  - すべての無効な定義スタンザおよび障害の原因。
  - すべての新規マシン・グループおよび追加するメンバー。
  - すべての既存マシン・グループおよび追加するメンバー。
  - NIM 環境に追加するすべてのネットワーク定義。
  - それぞれの新規マシンを追加するために呼び出すコマンド。
  - それぞれの追加マシン・インターフェースを追加するために呼び出すコマンド。
  - 新規マシン・グループを作成してそのメンバーを追加するために呼び出すコマンド。
  - 既存マシン・グループに新規メンバーを追加するために呼び出すコマンド。
- 注:** マシンの追加に使用する前にすべてのスタンザが正しいことを確認するため、クライアント定義ファイルで **-p** フラグを指定することをお勧めします。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0** 正常終了。
- 10** エラーが発生しました。

## セキュリティ

アクセス制御: このコマンドを実行するには、root 権限を持っていないければなりません。

イベントの監査: N/A

## 例

- クライアント定義ファイル **client.defs** をプレビューするには、次のように入力します。

```
nimdef -p -f client.defs
```
- クライアント定義ファイル **client.defs** で記述されている NIM クライアントを追加するには、次のように入力します。

```
nimdef -d -f client.defs
```
- client.add** という kshell スクリプトを作成して、クライアント定義ファイル **client.defs** で記述されている NIM クライアントを追加するには、次のように入力します。

```
nimdef -c -f client.defs > client.add
```

## ファイル

**/usr/sbin/nimdef** **nimdef** デーモン/コマンドが入っています。

## 関連情報

lsnim コマンド、nim コマンド、nimclient コマンド、nimconfig コマンド。

---

## niminit コマンド

### 目的

ネットワーク・インストール管理 (NIM) クライアント・パッケージを構成します。

### 構文

#### NIM クライアント・パッケージを構成する

```
niminit{-a name=Name -a pif_name=Pif -a master=Hostname} [ -a master_port=PortNumber ] [ -a registration_port=PortNumber ] [ -a cable_type=Type | -a ring_speed=Speed ] [-a iplrom_emu=Device ] [ -a platform=PlatformType ] [ -a netboot_kernel=NetbootKernelType ] [-a adpt_add=AdapterAddress] [ -a is_alternate= yes | no ] [ -a connect=value ]
```

#### /etc/niminfo ファイルを再作成する

```
niminit {-a name=Name -a master=Hostname -a master_port=PortNumber}
```

### 説明

**niminit** コマンドは、NIM クライアント・パッケージを構成します。この作業をしておかないと、**nimclient** コマンドは使用できません。**niminit** コマンドの必須属性を指定すると、**niminit** コマンドを実行しているマシンを表す新しいマシン・オブジェクトが作成されます。**niminit** コマンドが正常に終了すると、マシンは NIM 環境に入れるようになります。

NIM クライアント・パッケージを正常に構成した後、**niminit** コマンドを再び実行して、クライアント上で **/etc/niminfo** を再作成できます。**/etc/niminfo** ファイルは **nimclient** コマンドで使用するので、ユーザーが誤って除去してしまった場合は再作成しなければなりません。

このコマンドは、**is\_alternate** 属性が **yes** に設定されているときに、**alternate\_master** を構成します。**alternate\_master** を構成する前に、**bos.sysmgmt.nim.master** ファイルセットがインストールされている必要があります。**alternate\_master** の構成が正常に行われれば、登録されたマスターは **alternate\_master** 操作をこのマシン上で実行できるようになります。

### フラグ

<b>-a</b>	<b>niminit</b> コマンドに 5 つまでの属性を指定します。次の属性 = 値の組は、いずれも前に <b>-a</b> フラグが付いています。
<b>name=Name</b>	NIM がワークステーションの識別に使用する名前を指定します。この値は必須です。
<b>pif_name=Pif</b>	すべての NIM コミュニケーションのネットワーク・インターフェースの名前を定義します。この値は必須です。
<b>master=Hostname</b>	NIM マスターのホスト名を指定します。クライアントは、このホスト名をインターネット・プロトコル (IP) アドレスに対して解決できなければなりません。この値は必須です。

<b>master_port</b> = <i>PortNumber</i>	NIM コミュニケーションに使用する <b>nimesis</b> デーモンのポート番号を指定します。
<b>cable_type</b> = <i>CableType</i>	イーサネット・ケーブル・タイプを指定します。 <b>pif_name</b> がイーサネット・ネットワークを参照する場合は、この値が必要です。受け入れ可能な値は、 <b>bnc</b> 、 <b>dix</b> 、および <b>N/A</b> です。
<b>ring_speed</b> = <i>Speed</i>	Mbps 単位のスピード。 <b>pif_name</b> がトークンリング・ネットワークを参照する場合は、この値が必要です。受け入れ可能な値は、 <b>4</b> および <b>16</b> です。
<b>iplrom_emu</b> = <i>Device</i>	ROM エミュレーション・イメージが入っているデバイスを指定します。ネットワーク・インターフェース経由のブートが内部でサポートされないモデルには、このイメージが必須です。
<b>platform</b> = <i>PlatformType</i>	クライアントのマシン・タイプに対応するプラットフォームを指定します。この属性を指定しないと、デフォルトの <b>chrp</b> が使用されます。サポートされているプラットフォームは次のとおりです。
	<p><b>chrp</b> PowerPC Common Hardware Reference Platform (CHRP) アーキテクチャー・ベースのマシン</p> <p><b>rs6k</b> AIX 5.1 およびそれ以前用のマイクロチャネル・ベースのユニプロセッサ・モデル</p> <p><b>rs6ksmp</b> AIX 5.1 およびそれ以前用のマイクロチャネル・ベースの対称型マルチプロセッサ・モデル</p> <p><b>rspc</b> AIX 5.1 およびそれ以前用の PowerPC PCI バス・ベースのユニプロセッサ・マシン</p> <p><b>rspsmp</b> AIX 5.1 およびそれ以前用の PowerPC PCI バス・ベースの対称型マルチプロセッサ・マシン</p>
<b>adpt_add</b> = <i>AdapterAddress</i>	ネットワーク・アダプターに対応するハードウェア・アドレスを指定します。
<b>registration_port</b> = <i>PortNumber</i>	NIM クライアント登録に使用するポート番号を指定します。
	<p><b>注:</b></p> <ol style="list-style-type: none"> <li>1. コマンド行でポート番号を指定しない場合は、NIM の <b>/etc/services</b> ファイルのポート番号が使用されます。 <b>/etc/services</b> ファイルに NIM ポート <b>nim</b> および <b>nimreg</b> のエントリーがない場合、<b>master_port</b> では 1058、<b>registration_port</b> では 1059 のデフォルト値が使用されます。</li> <li>2. <b>master_port</b> および <b>registration_port</b> に使用する値は、NIM マスターが使用する値と同じものでなければなりません。NIM マスターが使用する値を表示するには、NIM マスター上でコマンド <b>lsnim -l master</b> を実行します。</li> </ol>
<b>netboot_kernel</b> = <i>NetbootKernelType</i>	ネットワーク上でクライアントをブートする際に使用するカーネルのタイプを指定します。 <b>netboot_kernel</b> の値は次のとおりです。
	<p><b>up</b> ユニプロセッサ・マシンのカーネル。</p> <p><b>mp</b> マルチプロセッサ・マシンのカーネル。</p> <p>デフォルト値は <b>up</b> です。</p>

**is\_alternate**=[yes|no]

このマシンを **alternate\_master** として構成する必要がある場合は、「yes」に設定します。

**connect**= *value*

NIM コマンドをリモート実行する場合に、NIM が使用する通信サービスを指定します。値のオプションは、shell (rsh の場合) および nimsh です。デフォルト設定は connect=shell です。この属性はオプションです。

## セキュリティ

アクセス制御: **nimit** コマンドを実行するには、root 権限が必要です。

### 例

1. BOOTP が使用可能になっている IPL ROM を持ち、NIM 環境では scuba として認識され、1 次インターフェースとして en0 を使用し、イーサネット・ケーブル・タイプとして bnc を指定し、マスターのホスト名 manta およびネットワーク・インストール・コミュニケーションの **/etc/services** にあるデフォルト NIM ポートを使用するようなマシン上で、NIM クライアント・パッケージを構成するには、次のように入力します。

```
nimit -a name=scuba -a pif_name=en0 -a cable_type=bnc \  
-a master=manta
```

2. **/etc/niminfo** ファイルがユーザーによって誤って除去された場合に、マスターのホスト名として superman のホスト名、ポート番号として 1058 を指定して **/etc/niminfo** ファイルを再作成するには、次のように入力します。

```
nimit -a name=robin -a master=superman -a master_port=1058
```

3. BOOTP 対応 IPL ROM を持つ PowerPC PCI バス・ベースのユニプロセッサ・システムであるマシン上で、AIX 5.1 およびそれ以前用の NIM クライアント・パッケージを、NIM 環境で starfish として認識されるように構成するには、次のように入力します。ただし、1 次インターフェースとして en0 を使用し、イーサネット・ケーブル・タイプは dix であり、マスターのホスト名 whale およびポート番号 1058 を使用して NIM マスターと通信するように指定します。

```
nimit -a name=starfish -a pif_name=en0 -a cable_type=dix \  
-a master=whale -a master_port=1058 -a platform=rspc
```

4. あるマシン上で NIM クライアントを構成するには、次のように入力します。その場合に、1 次インターフェースとしての at0 を使用し、マスターのホスト名 redfish およびポート番号 1058 を使用する NIM マスターと通信することを指定して、NIM 環境で bluefish として認識されるようにします。

```
nimit -a name=bluefish -a pif_name=at0 -a master=redfish \  
-a master_port=1058
```

**注:** ATM ネットワークへのインターフェースは、現在ネットワーク上のブートをサポートしていないので、この操作では、クライアントのサブネットに対応する汎用ネットワーク・オブジェクトが既に定義されている場合に NIM マスターのマシン・オブジェクトを定義します。

5. BOOTP 対応 IPL ROM を持つ PowerPC PCI バス・ベースの対称型マルチプロセッサ・システム・システムであるマシン上で、AIX 5.1 およびそれ以前用の NIM クライアント・パッケージを、NIM 環境で jellyfish として認識されるように構成するには、次のように入力します。ただし、1 次インターフェースとして en0 を使用し、イーサネット・ケーブル・タイプは dix であり、マスターのホスト名 whale およびポート番号 1058 を使用して NIM マスターと通信するように指定します。

```
nimit -a name=jellyfish -a pif_name=en0 -a cable_type=dix \  
-a master=whale -a master_port=1058 -a platform=rspcsm
```

6. デバイス `/dev/fd0` で IPL ROM エミュレーションを使用するあるマシン上で、NIM クライアント・パッケージを構成するには次のように入力します。その場合に、NIM 環境では `octopus` として認識され、1 次インターフェースとして `tr0`、リング・スピードとして `16` を使用し、マスターのホスト名 `dolphin` とクライアント・コミュニケーションにはポート番号 `1700`、クライアント登録には `1701` を使用する NIM マスターと通信するようにします。

```
niminit -a iplrom_emu=/dev/fd0 -a name=octopus -a pif_name=tr0 \  
-a ring_speed=16 -a master=dolphin -a master_port=1700 \  
-a registration_port=1701
```

7. このマシンを、NIM マスターが `dolphin` である **alternate\_master** として構成し、インターフェース `en0` を介して通信するようにするには、次のように入力します。

```
niminit -a is_alternate=yes -a name=octopus -a pif_name=en0 \  
-a cable_type=bnc -a master=dolphin
```

## ファイル

`/etc/niminfo` NIM が使用する変数が入っています。

## 関連情報

`lsnim` コマンド、`nim` コマンド、`nimclient` コマンド、`nimconfig` コマンド。

`.info` ファイル。

---

## niminv コマンド

### 目的

これによりシステム管理者は NIM オブジェクトのインストール・インベントリーに基づきフィックスを集め、集計し、比較し、ダウンロードできるようになります。

### 構文

インストール・インベントリーを取得する場合：

```
niminv -o invget -a targets=object1,object2,... [ -a location=path ] [ -a colonsep=yesno ]
```

インストール・インベントリーを集計する場合：

```
niminv -o invcon -a targets=object1,object2,... [ -a base=highest/lowest ] [ -a location=path ] [ -a colonsep=yesno ]
```

インストール・インベントリーを比較する場合：

```
niminv -o invcmp -a targets=object1,object2,... [ -a base=object|any ] [ -a location=path ]
```

集計インベントリーに基づきフィックスを取得する場合：

```
niminv -o fixget -a targets=object1,object2,... [ -a download=yesno ] [ -a lp_source=object ] [ -a location=path ] -a newlppname=name
```

## 説明

**niminv** (Network Install Manager Inventory) コマンドを使用すると、システム管理者は以下のタスクを完遂することができます。

- 種々の NIM オブジェクトのインストール・インベントリーを集める。
- 種々の NIM オブジェクトのインストール・インベントリーを集計する。
- 種々の NIM オブジェクトのインストール・インベントリーを比較する。
- 種々の NIM オブジェクトのインストール・インベントリーに基づきフィックスをダウンロードする。

**niminv** コマンドは、インストール情報を含んだ任意の NIM オブジェクトを使用することができます。かかるオブジェクトの例としては、 スタンドアロン・クライアント・オブジェクト、SPOT オブジェクト、**lpp\_source** オブジェクトおよび **mksysb** オブジェクトがあります。

**niminv** コマンドを使用する利点としては、以下のものがあります。

- ソフトウェア・インストール・インベントリーと並行して、ハードウェア・インストール・インベントリーが集められる。
- データ・ファイルが、容易に認識できる命名規則を使用して保管される。
- インストール・インベントリーを持つすべての NIM オブジェクトを使用できる。
- 当コマンドは、すべての管理対象 NIM オブジェクトの全体的なビューを提供する。

**niminv** によって表示される情報を、以下のファクターを用いて、限定することができます。

- 実際には物理デバイス (SPOT オブジェクト、**lpp\_source** オブジェクト、 **mksysb** オブジェクトなど) を持たないオブジェクトについて、ソフトウェア・インストール・インベントリーのみを提供する。
- クライアント・オブジェクトのソフトウェアおよびハードウェア・インストール・インベントリーについては、リモート・システム上のコマンドが提供可能なものに限定する。
- ダウンロードするフィックスの認識は、フィックス・バックエンド・サーバーに基づくものとする。詳細については、Using the Software Service Management menu (including SUMA) を参照してください。

## フラグ

**-a attribute=value** 属性と値を指定します。サポートされる属性と値は、その操作に基づきます。



**-o operation**

操作を指定します。以下に示す操作が現在、サポートされています。

**fixget** インストール・インベントリーに基づき、最新のフィックスを集めます。この操作は、以下の属性をサポートします。

**targets** (必須) フィックス収集の基本となる、コンマで区切られた NIM オブジェクトのリスト。

**lpp\_source**

(オプション) フィックスのダウンロード時にフィルターとして使用される、NIM **lpp\_source** オブジェクト。location と **newlppname** 属性が使用されていない場合は、この **lpp\_source** オブジェクトがフィックスのダウンロード先にもなります。

**location**

(オプション) フィックスを保管するディレクトリー。この属性は、**lpp\_source** 属性に与えたオブジェクトにフィックスをダウンロードしない場合にのみ使用します。この属性は、**newlppname** 属性と共にのみ使用されます。

**newlppname**

(オプション) 指定された場所 (location) で作成される **lpp\_source** の NIM オブジェクト名。この属性は、**location** 属性と共にのみ使用されます。指定する値は、明確で、しかも NIM 環境で現在、未使用のものでなければなりません。

**download**

(オプション) コマンドにフィックスをダウンロードさせるかどうかを指示します。 **lpp\_source** または **location** フィールドを指定せず、しかもこの属性の値が **yes** の場合は、フィックスは **suma** コマンドを介してデフォルトの場所にダウンロードされます。

**注:** **suma** コマンドは、**suma** 構成内の **MaxFSSize** フィールドに従って、ファイルシステム・スペースを増やします。

## invcmp

インストール・インベントリーを比較します。この操作は、以下の属性をサポートします。

**targets** (必須) インストール・インベントリーを比較する、コンマで区切られた NIM オブジェクトのリスト。

**base** (必須) 比較のベースとして使用する NIM オブジェクト、またはキーワード **any**。NIM オブジェクトを与えた場合、オブジェクト内のインストール・インベントリーが、表示されるデータの唯一の決定物となり、基本オブジェクト内のインベントリーのみがターゲット・オブジェクト内のインベントリーと比較されます。キーワード **any** は、コマンドに対してターゲットのインストール・インベントリーを強制的に使用させます。

## location

(オプション) データ・ファイルを保管するディレクトリー。このオプションを使用すると、各インベントリーはフォーマット

**conglomerate***base\_object, target\_object\_list.timestamp* で保管されま  
す。ここで、*base\_object* は、比較で使用される基本オブジェクトの  
NIM 名 (またはキーワード **any**) であり、*target\_object\_list* は、オ  
ブジェクトの NIM 名をコロンで区切り、ソートしたリストであ  
り、また *timestamp* は、コマンドが実行された時間 (年 月 日 時  
分 秒) です。ディレクトリーが存在しない場合は、それが作成さ  
れます。デフォルトは、データの画面への表示です。

-o operation (続き)

- invcon** インストール・インベントリーを集計します。この操作は、以下の属性をサポートします。
- targets** (必須) インストール・インベントリーを集計するための、NIM オブジェクトのコマで区切られたリスト。
- base** (オプション) 集計インベントリーが最高のソフトウェア・レベル、または最低のソフトウェア・レベルのいずれに基づくかを指定します。
- location**  
(オプション) データ・ファイルを保管するディレクトリー。このオプションを使用すると、各インベントリーはフォーマット **base.target\_object\_list.timestamp** で保管されます。ここで、**base** は、集計が最高レベルまたは最低レベルのいずれに基づくか、**target\_object\_list** は、オブジェクトの NIM 名がコロンで区切られ、ソートされたリスト、また **timestamp** は、コマンドが実行された時間 (年 月 日 時 分 秒) です。ディレクトリーが存在しない場合は、それが作成されます。デフォルトは、データの画面への表示です。
- colonsep**  
(オプション) コマンドにコロンで区切られた出力を作成させるかどうかを指示します。デフォルトは **no** です。
- invget** インストール・インベントリーを集めます。この操作は、以下の属性をサポートします。
- targets** (必須) インストール・インベントリーを集めるための、コマで区切られた NIM オブジェクトのリスト。
- location**  
(オプション) データ・ファイルを保管するディレクトリー。このオプションを使用すると、各インベントリーはフォーマット **conglomerate.target\_object\_name.timestamp** で保管されます。ここで、**target\_object\_name** は、オブジェクトの NIM 名、**timestamp** は、コマンドが実行された時間 (年 月 日 時 分 秒) です。ディレクトリーが存在しない場合は、それが作成されます。デフォルトは、データの画面への表示です。
- colonsep**  
(オプション) コマンドにコロンで区切られた出力を作成させるかどうかを指示します。デフォルトは **no** です。

## 終了状況

- 0 コマンドは正常に実行されました。  
>0 エラーが発生しました。

## 例

- 2 つのクライアントのインストール・インベントリーを集めて、出力を **/tmp/inventory** に保管するには、以下のように入力します。

```
niminiv -o invget -a targets=client1,client2 -a location=/tmp/inventory
```

以下のような出力が表示されます。

```
Installation Inventory for client1 saved to
/tmp/inventory/inventory.client1.060406140453.
Installation Inventory for client2 saved to
/tmp/inventory/inventory.client2.060406140453.
```

ファイル内の情報は **lspp -L** の出力に似ています。

- 2 つのクライアントのインストール・インベントリを集計して、出力を **/tmp/inventory** に保管するには、以下のように入力します。

```
niminiv -o invcon -a targets=client1,client2 -a location=/tmp/inventory
```

以下のような出力が表示されます。

```
Installation Inventory for client1 saved to
/tmp/inventory/conglomerate.client1:client2.060406140500.
```

ファイル内の情報は **lspp -L** の出力に似ています。

3. **mksysb**、**SPOT**、および **lpp\_source** のインストール・インベントリを、**master** に現在インストールされているものと比較し、出力を **/tmp/inventory** に保管するには、以下のように入力します。

```
niminiv -o invcon -a targets=mksysb1,spot1,lpp_source1 -a base=master -a \
location=/tmp/inventory
```

以下のような出力が表示されます。

```
Installation Inventory for client1 saved to
/tmp/inventory/comparison.master.mksysb1:spot1:lpp_source1.060406140610.
```

ファイル内の情報は列形式でリストされます。比較に含まれるのは、**master** 上のインストール・インベントリのみです。

4. 先の例と同じ比較を行い、しかも任意のオブジェクトのソフトウェアも含める場合は、以下のように入力します。

```
niminiv -o invcon -a targets=mksysb1,spot1,lpp_source1,master -a base=any -a \
location=/tmp/inventory
```

以下のような出力が表示されます。

```
Installation Inventory for client1 saved to
/tmp/inventory/comparison.any.mksysb1:spot1:lpp_source1.060406140733.
```

ファイル内の情報は列形式でリストされます。比較には、任意のターゲット・オブジェクト内のすべてのインストール・インベントリが含まれます。

5. **mksysb**、**SPOT** および **lpp\_source** 内の最低のインストールに基づきダウンロード可能なフィックスを見つけるには、以下のように入力します。

```
niminiv -o fixget -a targets=mksysb1,spot1,lpp_source1
```

以下のような出力が表示されます。

```
*****
Performing preview download.
*****
```

```
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/Java14.debug.1.4.1.0.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/Java14.debug.1.4.1.7.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/devices.pci.00100100.com.5.2.0.50.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/devices.pci.00100300.diag.5.2.0.75.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/devices.pci.00100f00.rte.5.2.0.85.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/devices.pci.13100560.rte.5.2.0.85.bff
```

```
Summary:
  6 downloaded
  0 failed
  0 skipped
```

6. **mksysb**、**SPOT** および **lpp\_source** 内の最低のインストールに基づき最新のフィックスをダウンロードするには、以下のように入力します。

```
niminv -o fixget -a targets=mksysb1,spot1,lpp_source1 -a download=yes
```

以下のような出力が表示されます。

```
Extending the /usr filesystem by 30 blocks.
File System size changed to 8126464
```

```
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/Java14.debug.1.4.1.0.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/Java14.debug.1.4.1.7.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/devices.pci.00100100.com.5.2.0.50.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/devices.pci.00100300.diag.5.2.0.75.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/devices.pci.00100f00.rte.5.2.0.85.bff
Download SUCCEEDED: /usr/sys/inst.images/installp/ppc/devices.pci.13100560.rte.5.2.0.85.bff
```

```
Summary:
  6 downloaded
  0 failed
  0 skipped
```

**注:** (**suma** コマンドで指定され) デフォルトのダウンロード・パスに既に含まれているインストールは、再度ダウンロードされることはありません。この例でのデフォルトのダウンロード・パスは、**/usr/sys/inst.images** です。デフォルトのダウンロード・パスの存在場所に関する詳細については、**suma** コマンドを参照してください。

7. **mksysb**、**SPOT** および **lpp\_source** 内の最低のインストールに基づき最新のフィックスを既存の **lpp\_source** にダウンロードするには、以下のように入力します。

```
niminv -o fixget -a targets=mksysb1,spot1,lpp_source1 -a download=yes -a \
lpp_source=lpp_source2
```

以下のような出力が表示されます。

```
Download SUCCEEDED: /nim/lpps/lpp_source2/installp/ppc/Java14.debug.1.4.1.0.bff
Download SUCCEEDED: /nim/lpps/lpp_source2/installp/ppc/Java14.debug.1.4.1.7.bff
```

```
Summary:
  2 downloaded
  0 failed
  0 skipped
```

**注:** 既に **lpp\_source2** に含まれているインストールは再度ダウンロードされることはありません。この例では、**filesets** デバイスは **lpp\_source2** に既に存在しています。

8. **mksysb**、**SPOT** および **lpp\_source** 内の最低のインストールに基づき最新のフィックスを新しい **lpp\_source** にダウンロードするには、以下のように入力します (ただし、フィルター処理ファイルセットは既存の **lpp\_source** にあります)。

```
niminv -o fixget -a targets=mksysb1,spot1,lpp_source1 -a download=yes -a \
location=/nim/lpps/newlpp1 -a newlppname=newlpp1
```

以下のような出力が表示されます。

```
Download SUCCEEDED: /nim/lpps/newlpp1/installp/ppc/Java14.debug.1.4.1.0.bff
Download SUCCEEDED: /nim/lpps/newlpp1/installp/ppc/Java14.debug.1.4.1.7.bff
```

```
Summary:
  2 downloaded
  0 failed
  0 skipped
```

注: 既に **lpp\_source2** に含まれているインストールは再度ダウンロードされることはありません。この例では、**filesets** デバイスは **lpp\_source2** に既に存在しています。

9. **mksysb**、**SPOT** および **lpp\_source** 内の最低のインストールに基づき最新のフィックスを新規の **lpp\_source** にダウンロードするには、以下のように入力します。

```
niminv -o fixget -a targets=mksysb1,spot1,lpp_source1 -a download=yes -a \
location=/nim/lpps/newlpp2 -a newlppname=newlpp2
```

以下のような出力が表示されます。

```
Download SUCCEEDED: /nim/lpps/newlpp2/installp/ppc/Java14.debug.1.4.1.0.bff
Download SUCCEEDED: /nim/lpps/newlpp2/installp/ppc/Java14.debug.1.4.1.7.bff
Download SUCCEEDED: /nim/lpps/newlpp2/installp/ppc/devices.pci.00100100.com.5.2.0.50.bff
Download SUCCEEDED: /nim/lpps/newlpp2/installp/ppc/devices.pci.00100300.diag.5.2.0.75.bff
Download SUCCEEDED: /nim/lpps/newlpp2/installp/ppc/devices.pci.00100f00.rte.5.2.0.85.bff
Download SUCCEEDED: /nim/lpps/newlpp2/installp/ppc/devices.pci.13100560.rte.5.2.0.85.bff
```

```
Summary:
  6 downloaded
  0 failed
  0 skipped
```

## 位置

**/usr/sbin/niminv**

## 関連情報

**installp Command**、**lspp Command**、**lsmcode Command**、**lscfg Command**、91 ページの『**nim コマンド**』、**suma Command** のセクション。

---

## nimol\_backup コマンド

### 目的

AIX クライアントから NIMOL インストール・リソースを作成します。

### 構文

```
nimol_backup -c client_hostname [-t directory] [-m remote_access_method] [-L label] [-D]
```

### 説明

**nimol\_backup** コマンドは、クライアント上で **nimol\_mk\_resources** 方式を呼び出すために、指定されたりモート・アクセス方式 (デフォルトでは **/usr/bin/rsh**) を使用して、構成済みの NIMOL クライアントから NIMOL インストール・リソースを作成します。 **nimol\_config** コマンドを使用して NIMOL サーバーを構成する場合、ユーザーはデフォルトのリモート・アクセス方式を、**/usr/bin/rsh** 以外の例えば **/usr/bin/ssh** などに設定することができます。 **-n** フラグなしに **nimol\_install** コマンドを使用してインストールされたマシンの場合は、NIMOL クライアントとして認識されます。

このコマンドは、ターゲット・ディレクトリーとラベルを NIMOL サーバー上に作成します。そしてディレクトリーがエクスポートされます。デフォルト・ラベルは `default` です。例えば、コマンドが `-t /export/aix -L aix530` を渡されると、コマンドは、`/export/aix/aix530` ディレクトリーを NIMOL サーバー上に作成します。

それから、コマンドは、リモート・アクセス方式を使用して `nimol_mk_resources` コマンドを実行します。`nimol_mk_resources` コマンドは、必要なインストール・リソースをターゲット・ディレクトリー内に作成します。

## フラグ

<code>-c client_hostname</code>	<b>geninstall</b> コマンドを実行する場所となる、NIMOL クライアント・ホスト名を指定します。
<code>-D</code>	コマンドをデバッグ・モードで実行します。
<code>-L label</code>	作成されたリソース用に作成する、ラベルまたは名前を指定します。
<code>-m remote_access_method</code>	<b>geninstall</b> コマンドを実行するために使用する、リモート・アクセス方式を指定します。デフォルトは <code>/usr/bin/rsh</code> です。もう 1 つのオプションは、 <code>/usr/bin/ssh</code> です。
<code>-t directory</code>	NIMOL クライアントから AIX インストール・リソースが作成される場所となる、ターゲット・ディレクトリーを指定します。デフォルト・ディレクトリーは <code>/export/aix</code> です。

## 終了状況

0	コマンドは正常に実行されました。
> 0	エラーが戻されました。

## セキュリティ

`nimol_backup` コマンドを NIMOL クライアント上で実行するには、クライアントは NIMOL サーバーに対してリモート・アクセス許可を与える必要があります。`/usr/bin/ssh` の使用は、`/usr/bin/rsh` よりさらに機密保護機能のあるリモート・アクセス方式です。

## 例

1. `/export/aix` ディレクトリー内のクライアント `myclient` から、530 という名前でインストール・リソースを作成するには、次のように入力します。

```
nimol_backup -c myclient -L 530 -t /export/aix
```

2. `ssh` を使用して `nimol_mk_resources` を実行するには、次のように入力します。

```
nimol_backup -c myclient -m ssh
```

## 位置

`/usr/sbin/nimol_backup`

## ファイル

`/etc/nimol.conf` コマンドの構成情報を保管します。



## 関連情報

『nimol\_config コマンド』、165 ページの『nimol\_install コマンド』、167 ページの『nimol\_lslpp コマンド』、169 ページの『nimol\_update コマンド』。

---

## nimol\_config コマンド

### 目的

サービスを構成しインストール・リソースをコピーして AIX をマシンにネットワーク・インストールするために、Linux サーバーを構成します。

### 構文

**nimol\_config** [-d *DirectoryContainingAIXResources*] [-t *TargetDirectoryToCopyResources*] [-L *InstallResourcesLabel*] [-s *NIMOLServerHostname*] [-m *RemoteAccessMethod*] [-C] [-e] [-l] [-r] [-S] [-U] [-D]

### 説明

**nimol\_config** コマンドは、AIX をマシンにネットワーク・インストールするために、Linux サーバーを構成します。このコマンドは、次の構成を実行します。

1. コマンドは、まず、Linux サーバーのホスト名と IP アドレスを取得します。 **-s** フラグとともにホスト名が指定されていない場合は、コマンドはローカル・マシンのホスト名およびそのホスト名に関連付けられた IP アドレスを使用します。ホスト名と IP アドレスが指定されている場合は、この対がまだ **/etc/hosts** ファイルに存在していなければ追加されます。
2. コマンドは次に、ポート・マップ・サービスおよび NFS サーバーを開始します。
3. **-m** フラグがともに指定されている場合は、コマンドはリモート・アクセス方式を **/etc/nimol.conf** ファイルに保管します。デフォルトのリモート・アクセス・コマンドは **/usr/bin/rsh** で、これは **nimol\_install** コマンドに対して **-n** フラグの指定なしにインストールされた NIMOL クライアントと通信するために使用されます。
4. 次に、tftpboot が構成されます。**/tftpboot** ディレクトリが存在しなければ作成され、**/etc/xinetd.d/tftp** ファイルが存在しなければ作成されます。そして、コマンドは **/etc/xinetd.d/tftp** ファイル内で **no** と等しい使用不可に設定し、tftp サーバーが着信要求を処理できるように xinetd を再始動します。
5. **nimol\_config** コマンドは、他のマシンからの着信メッセージを受信するために、SYSLOG のセットアップも行います。クライアントは、SYSLOG サーバーをインストールする際に状況を渡り返します。**/etc/sysconfig/syslog** ファイルは、SYSLOGD\_OPTIONS 変数または SYSLOGD\_PARAMS 変数に **-r** フラグを組み込むように変更されます。そしてコマンドは、使用可能な最初のローカル・ログを求めて **/etc/syslog.conf** を検索し、**/var/log/nimol.log** にメッセージを書き込むようにセットします。クライアントはこのログ・ファイルに状況を書き込みます。このファイルはクライアント・インストール時にモニターすることができます。SYSLOG 構成ファイルへの変更が行われた後、サービスが再始動されます。
6. 次にコマンドは、AIX クライアントからの bootp 要求を受け取るように DHCP サーバーをセットアップします。NIMOL サーバーのサブネットが決定され、**dhcpcd.conf** ファイルに追加されます。オプション **allow bootp**、**not authoritative**、および **ddns-update-style none** がまだ存在していなければ、追加されます。これらのオプションの既存の設定値は上書きされることとなります。
7. サービスが構成されると、**nimol\_config** コマンドは、**-C** フラグがコマンドに渡されていない場合は、AIX インストール・リソースをローカルにコピーしようとします。コマンドは、**-d** フラグで

指定されたソース・ディレクトリー (デフォルトでは **/mnt/cdrom**) からターゲット・ディレクトリー (デフォルトでは **/export/aix**) へ、リソースをコピーします。ディレクトリーが作成されます (**-L** フラグに指定された LABEL 名と一致する名前、デフォルトは「default」)。コマンドは次のリソースのソース・ディレクトリーの中を調べます。

- **/SPOT** という名前の SPOT (Source Product Object Tree) ディレクトリー、および **ispot.tar.Z** という名前の SPOT ディレクトリー
- **/lpp\_source** という名前の lpp\_source ディレクトリー
- **mksysb** または **mksysb.bff** という名前の mksysb
- **booti.chrp.mp.ent** という名前のブート・イメージ
- **bosinst.data** という名前の bosinst.data ファイル
- **image.data** という名前の image.data ファイル
- **cust.script** という名前のカスタマイズ・スクリプト
- **resolv.conf** という名前の resolv.conf ファイル

SPOT、ブート・イメージ、および mksysb または lpp\_source のいずれかが必要です。

8. **-e** フラグが指定されている場合を除き、ターゲット・ディレクトリーがグローバルにエクスポートされます。
9. リソースを含むターゲット・ディレクトリーとラベルが指定されている場合、これらのリソースが使用され、リソースがコピーされることはありません。例えば、コマンドが **-t /export/aix -L aix530** を渡され、ディレクトリー **/export/aix/aix530** がリソースを含む場合、コマンドはソース・ディレクトリーからリソースをコピーする試みをしません。
10. NIMOL サーバーが構成された後、**nimol\_config** コマンドは、新しいリソース・ラベルを定義するときに NIMOL サーバー上でサービスを再構成する試みをしません。
11. また、コマンドは、**-i** フラグを持つ定義済みリソース・ラベルをリストします。
12. リソース・ラベルに **-r** フラグを指定することにより、リソース・ラベルを除去することができます。コマンドは、ディレクトリーがエクスポートされていればアンエクスポートし、リソース・ラベルのディレクトリーを削除します。
13. **-U** フラグが渡されると、コマンドは、構成解除サービスなど、行われた構成があれば元に戻す試みをします。

## フラグ

- |                            |   |
|----------------------------|---|
| <b>-C</b>                  | サーバーが、インストール・リソースをコピーせずにサービスを構成することのみ必要であることを指定します。   |
| <b>-d</b> <i>directory</i> | AIX インストール・ディレクトリーを含むソース・ディレクトリーを指定します。デフォルト・ディレクトリーは <b>/mnt/cdrom</b> です。                     |
| <b>-D</b>                  | コマンドをデバッグ・モードで実行します。  |
| <b>-e</b>                  | 新しく作成されたリソース・ラベルのディレクトリーをグローバルにエクスポートしないよう、コマンドに指示します。  |
| <b>-i</b>                  | クライアントをインストールするために使用可能な、定義済みリソース・ラベルをリストします。  |
| <b>-L</b> <i>label</i>     | コピーされたリソース用に作成する、ラベルまたは名前を指定します。  |
| <b>-m</b> <i>method</i>    | <b>nimol_install</b> コマンドに対して <b>-n</b> フラグの指定なしにインストールされたクライアントと通信するときに使用する、リモート・アクセス方式を指定します。 |
| <b>-r</b>                  | 指定されたリソース・ラベルを除去するよう、コマンドに指示します。  |
| <b>-s</b> <i>hostname</i>  | NIMOL サーバーに使用するホスト名。デフォルトでは、 <b>hostname</b> コマンドを実行することによりホスト名を決定します。                          |

- S** SYSLOG サービスを構成しないよう、コマンドに指示します。クライアントがインストールされるとき、状況はログに記録されません。
- t *directory*** AIX インストール・リソースがソース・ディレクトリーからコピーされるターゲット・ディレクトリーを指定します。デフォルト・ディレクトリーは **/export/aix** です。
- U** NIMOL サーバーを構成解除するよう、コマンドに指示します。コマンドは、自身が実行した構成があれば元に戻すを試みます。

## 終了状況

- 0** コマンドは正常に実行されました。
- > 0** エラーが戻されました。

## セキュリティ

リモート・クライアントからのメッセージを受信するために **SYSLOG** サービスを構成することは、セキュリティ上の問題となる場合があります。既知のクライアントからの **SYSLOG** メッセージのみを受信するよう、ファイアウォールを構成します。

## 例

1. リソースをコピーせずに **NIMOL** サーバーを構成するには、次のように入力します。  
`nimol_config -C`
2. **NIMOL** サーバーを構成し、**/mnt/aix** から **/export/aix** へリソースをコピーし、リソース **aix530** をラベル付けするには、次のように入力します。  
`nimol_config -d /mnt/aix -t /export/aix -L aix530`
3. **SYSLOG** を構成せず、リソース・ラベル・ディレクトリーをグローバルにエクスポートせずに、**NIMOL** サーバーを構成しリソースをコピーするには、次のように入力します。  
`nimol_config -S -e`
4. 定義済みリソース・ラベルをリストするには、次のように入力します。  
`nimol_config -l`
5. **aix530** リソース・ラベルを除去するには、次のように入力します。  
`nimol_config -L aix530 -r`

## 位置

**/usr/sbin/nimol\_config**

## ファイル

**/etc/nimol.conf** コマンドの構成情報を保管します。

## 関連情報

165 ページの『**nimol\_install** コマンド』, 167 ページの『**nimol\_lslpp** コマンド』, 169 ページの『**nimol\_update** コマンド』.

---

## nimol\_install コマンド

### 目的

AIX を特定のクライアント・マシンにインストールするための構成済み NIMOL サーバーをセットアップします。

### 構文

```
nimol_install -c client_hostname [ -g gateway ] [-m mac_address] [ -p ip_address ] [ -s subnet_mask ]  
[-L label] [ -n ] [ -r ] [-D]
```

### 説明

**nimol\_install** コマンドは、AIX 搭載マシンをネットワーク・インストールするために構成済み NIMOL サーバーをセットアップします。このコマンドは、次の構成を実行します。

1. コマンドは、クライアント IP アドレスが指定されていなければ、クライアント・ホスト名の IP アドレスを決定します。クライアント・ホスト名が解決可能でなく、クライアント IP アドレス (ドット 10 進数) が指定されている場合、この対が存在していなければ **/etc/hosts** ファイルに追加されます。
2. クライアントが **/etc/nimol.conf** ファイルに追加されます。
3. リソース・ラベルのディレクトリーが、まだグローバルにエクスポートされていなければ、クライアントにエクスポートされます。
4. クライアントのスタンザが、**/etc/dhcpd.conf** ファイルに追加されます。クライアントのサブネットも、もし存在していなければ **/etc/dhcpd.conf** ファイルに追加されます。クライアントまたはクライアントのサブネットが既に **/etc/dhcpd.conf** ファイルに存在する場合には、エラーが表示されます。
5. ブート・イメージへのシンボリック・リンクが、クライアント用に、**/ftftpboot** ディレクトリーに作成されます。
6. クライアントが NIMOL サーバーと同じサブネット上にある場合は、静的 arp エントリーが追加されます。
7. コマンドは、**iptables** コマンドが存在する場合、次のように実行することにより、インストール中のクライアントに対するファイアウォール・ルールをオフにします。

```
iptables -I INPUT 1 -s client_hostname -j ACCEPT
```

これにより、NIMOL が使用するさまざまなサービスが成功します。クライアントが除去される場合は、**nimol\_install** コマンドは、規則を削除するために次のコマンドを実行します。iptables -D INPUT -s *client\_hostname*

8. コマンドは、必要なリソースがリソース・ラベルのディレクトリーに存在していることを確認します。
9. **resolv.conf** またはカスタマイズ・スクリプトが指定された場合、またはクライアントがインストール後に NIMOL サーバーのクライアントのままである場合は、リソース・ラベルのディレクトリーのスクリプト・サブディレクトリーに **nim\_script** が作成されます。**nimol\_install** コマンドは、**cust.script** という名前のリソース・ラベルのディレクトリー内、または **client\_name.script** という名前のクライアント用の特定のカスタマイズ・スクリプト内を、汎用カスタマイズ・スクリプトを求めて探索します。
10. 情報ファイルが、オペレーティング・システムのインストール時に使用されることになる **/ftftpboot** ディレクトリー内に作成されます。

11. **-i** フラグが指定されている場合、コマンドは、インストール用にセットアップされたクライアントをリストします。クライアント名とともに **-r** フラグが指定されている場合は、クライアントは除去されます。
12. クライアントがインストール用にセットアップされたら、ネットワーク・インストールを実行するよう、クライアントに伝える必要があります。クライアントに AIX がインストールされており稼働中である場合は、**bootlist** コマンドを使用します。例えば、NIMOL サーバーが 192.168.1.20 で、AIX クライアントが 192.168.1.30 の場合、ent0 をブートオフするには、以下を実行します。

```
bootlist -m normal -ent0 bserver=192.168.1.20 \\  
gateway=0.0.0.0 client=192.168.1.30
```

そして、次を実行することによりリブートします。

```
shutdown -Fr
```

13. クライアントが稼働中でない場合は、SMS メニューにブートしてネットワーク・ブート・パラメーターとネットワーク・ブート・デバイスを指定します。クライアントが NIMOL サーバーと同じサブネット上にある場合は、クライアントはブロードキャスト **bootp** インストールを行うことができます。ブロードキャスト **bootp** は IP パラメーターがセットされている必要はありません。bserver、ゲートウェイ、およびクライアントは、ブロードキャスト **bootp** インストールにおいて 0.0.0.0 となります。

## フラグ

<b>-c</b> <i>client_hostname</i>	インストール用にセットアップされるクライアント・ホスト名、または除去されるクライアント・ホスト名を、指定します。
<b>-D</b>	コマンドをデバッグ・モードで実行します。
<b>-g</b> <i>gateway</i>	クライアントが AIX をインストールした後に構成されるゲートウェイを指定します。これは、クライアントをインストールしているときには必要です。
<b>-i</b>	インストール用にセットアップされるクライアントをリストします。
<b>-L</b> <i>label</i>	クライアントをインストールするのに使用されるリソースのラベルまたは名前を指定します。デフォルトは <code>default</code> です。
<b>-m</b> <i>mac_address</i>	クライアントがインストールされるネットワーク・インターフェースの MAC アドレスを指定します。これは、クライアントをインストールしているときには必要です。MAC アドレスにはコロンを含める必要があります (例えば、 <code>00:60:08:3F:E8:DF</code> )。
<b>-n</b>	インストールが完了した後に NIMOL サーバーのクライアントのままであるために、マシンを構成しないよう指定します。このオプションが指定されると、クライアントはインストール後に自身のネットワークを構成させることはありません。
<b>-p</b> <i>ip_address</i>	クライアントの IP アドレスを指定します。クライアントのホスト名が解決可能でない場合に、このフラグを使用します。
<b>-r</b>	クライアントを除去します。クライアントは、再構成されるまで、AIX をインストールすることはできません。このフラグにはクライアント・ホスト名が必要です。
<b>-s</b> <i>subnet_mask</i>	クライアント・インターフェースのサブネット・マスクを指定します。このフラグは、クライアントをインストールしているときは必須です。

## 終了状況

0	コマンドは正常に実行されました。
> 0	エラーが戻されました。

## セキュリティ

マシンが NIMOL サーバーのクライアントのままている場合 (**-n** フラグが指定されない場合)、NIMOL サーバーに **/usr/bin/rsh** 許可を与えることになり、サーバーはクライアント上でコマンドを実行することができます。

## 例

1. クライアント `myclient` をセットアップして、ゲートウェイ `192.168.1.1`、MAC アドレス `00:60:08:3F:E8:DF`、およびサブネット・マスク `255.255.255.0` を持つ `aix530` リソース・ラベルをインストールするには、次のように入力します。

```
nimol_install -c myclient -g 192.168.1.1 \<\  
-m 00:60:08:3F:E8:DF -s 255.255.255.0 -L aix530
```

2. クライアント `myclient` をセットアップし、インストール後は NIMOL サーバーに対するクライアントのままにはしないようにするには、次のように入力します。

```
nimol_install -n -c myclient -g 192.168.1.1 \<\  
-m 00:60:08:3F:E8:DF -s 255.255.255.0 -L aix530
```

3. インストールするために構成されたクライアントをリストするには、次のように入力します。

```
nimol_install -l
```

4. `myclient` を除去するには、次のように入力します。

```
nimol_config -c myclient -r
```

## 位置

**/usr/sbin/nimol\_install**

## ファイル

**/etc/nimol.conf**                    コマンドの構成情報を保管します。

## 関連情報

162 ページの『`nimol_config` コマンド』、『`nimol_lslpp` コマンド』、169 ページの『`nimol_update` コマンド』。

---

## nimol\_lslpp コマンド

### 目的

`lslpp` コマンドを NIMOL クライアント上で実行します。

### 構文

```
nimol_lslpp -c client_hostname [ -m remote_access_method ] [-f lslpp_flags ] [ -D ]
```

### 説明

`nimol_lslpp` コマンドは、指定されたりモート・アクセス方式 (デフォルトでは **/usr/bin/rsh**) を使用して、構成済みの NIMOL クライアント上で `lslpp` コマンドを実行します。 `nimol_config` コマンドを使用して NIMOL サーバーを構成する場合、ユーザーはデフォルトのリモート・アクセス方式を、**/usr/bin/rsh**



以外の例えば `/usr/bin/ssh` などに設定することができます。 `-n` フラグなしに `nimol_install` コマンドを使用してインストールされたマシンの場合は、NIMOL クライアントとして認識されます。

コマンドは、`-L -c` をデフォルト・フラグとして `lspp` コマンドを実行します。 `lspp` コマンド・フラグは、`-f` フラグとともに指定することができます。

## フラグ

<code>-c client_hostname</code>	<code>lspp</code> コマンドを実行する場所となる、NIMOL クライアント・ホスト名を指定します。
<code>-D</code>	コマンドをデバッグ・モードで実行します。
<code>-f lspp_flags</code>	<code>lspp</code> コマンドに渡す、 <code>lspp</code> コマンド・フラグを指定します。
<code>-m remote_access_method</code>	<code>lspp</code> コマンドを実行するために使用する、リモート・アクセス方式を指定します。デフォルトは <code>/usr/bin/rsh</code> です。もう 1 つのオプションは、 <code>/usr/bin/ssh</code> です。

## 終了状況

0	コマンドは正常に実行されました。
> 0	エラーが戻されました。

## セキュリティ

`nimol_lspp` コマンドを NIMOL クライアント上で実行するには、クライアントは NIMOL サーバーに対してリモート・アクセス許可を与える必要があります。 `/usr/bin/ssh` の使用は、`/usr/bin/rsh` よりさらに機密保護機能のあるリモート・アクセス方式です。

## 例

1. `lspp` コマンドをクライアント `myclient` で、デフォルト・フラグ `-Lc` を使用して実行するには、次のように入力します。

```
nimol_lspp -c myclient
```

2. `lspp` コマンドをクライアント `myclient` で、フラグ `-i bos.rte` を使用して実行するには、次のように入力します。

```
nimol_lspp -c myclient -f "-i bos.rte"
```

3. `lspp` コマンドをクライアント `myclient` で、リモート・アクセス方式として `ssh` を使用して実行するには、次のように入力します。

```
nimol_lspp -c myclient -m ssh
```

## 位置

`/usr/sbin/nimol_lspp`

## ファイル

`/etc/nimol.conf` コマンドの構成情報を保管します。



## 関連情報

162 ページの『nimol\_config コマンド』、165 ページの『nimol\_install コマンド』、『nimol\_update コマンド』。

---

## nimol\_update コマンド

### 目的

NIMOL クライアントで `geninstall` を実行して、ソフトウェア保守を行います。

### 構文

```
nimol_update -c client_hostname [ -L label ] [ -f geninstall_flags ] [ -m remote_access_method ] [ -p package_list ] [-D]
```

### 説明

**nimol\_update** コマンドは、指定されたリモート・アクセス方式 (デフォルトでは `/usr/bin/rsh`) を使用して、構成済みの NIMOL クライアント上で **geninstall** コマンドを実行します。**nimol\_config** コマンドを使用して NIMOL サーバーを構成する場合、ユーザーはデフォルトのリモート・アクセス方式を、`/usr/bin/rsh` 以外の例えば `/usr/bin/ssh` などに設定することができます。**-n** フラグなしに **nimol\_install** コマンドを使用してインストールされたマシンの場合は、NIMOL クライアントとして認識されます。

コマンドは、**-acgX** をデフォルト・フラグとして **geninstall** コマンドを実行します。**-f** フラグを使用して **geninstall** コマンド・フラグを指定します。**geninstall** コマンドを渡すソフトウェア・パッケージは、**-p** フラグで指定されます。

**nimol\_update** コマンドを使用してファイルセットをインストールする場合、`lpp_source` を持つリソース・ラベルを指定する必要があります。リソース・ラベルが `lpp_source` を含んでいるか判断するには、**nimol\_config -l -L label** を実行します。リソース・ラベル・ディレクトリーがまだグローバルにエクスポートされていない場合は、コマンドはリソース・ラベル・ディレクトリーをエクスポートします。クライアントはディレクトリーをマウントし、インストール時にソース・ディレクトリーとして使用します。

### フラグ

<b>-c</b> <i>client_hostname</i>	<b>geninstall</b> コマンドを実行する場所となる、NIMOL クライアント・ホスト名を指定します。
<b>-D</b>	コマンドをデバッグ・モードで実行します。
<b>-f</b> <i>geninstall_flags</i>	<b>geninstall</b> コマンドに渡すフラグを指定します。デフォルト・フラグは、 <b>acgX</b> です。
<b>-L</b> <i>label</i>	インストール・イメージのソースとして使用されるリソース・ラベルの名前を指定します。
<b>-m</b> <i>remote_access_method</i>	<b>geninstall</b> コマンドを実行するために使用する、リモート・アクセス方式を指定します。デフォルトは <code>/usr/bin/rsh</code> です。もう 1 つのオプションは、 <code>/usr/bin/ssh</code> です。
<b>-p</b> <i>package_list</i>	<b>geninstall</b> コマンドに渡すソフトウェア・パッケージの名前を指定します。デフォルトは <code>all</code> です。

## 終了状況

0                    コマンドは正常に実行されました。  
> 0                   エラーが戻されました。

## セキュリティ

**nimol\_update** コマンドを NIMOL クライアント上で実行するには、クライアントは NIMOL サーバーに対してリモート・アクセス許可を与える必要があります。 **/usr/bin/ssh** の使用は、**/usr/bin/rsh** よりさらに機密保護機能のあるリモート・アクセス方式です。

## 例

1. リソース・ラベル 530 のすべてのパッケージをクライアント **myclient** にインストールするには、次のように入力します。

```
nimol_update -c myclient -L 530
```

2. クライアント **myclient** 上の **bos.games** に更新を適用するには、次のように入力します。

```
nimol_update -c myclient -L 530 -f "-a" -p "bos.games"
```

3. **bos.games** をクライアント **myclient** から除去するには、次のように入力します。

```
nimol_update -c myclient -f "-u" -p "bos.games"
```

4. **ssh** を使用して **geninstall** コマンドを実行するには、次のように入力します。

```
nimol_update -c myclient -L 530 -m ssh
```

## 位置

**/usr/sbin/nimol\_update**

## ファイル

**/etc/nimol.conf**                    コマンドの構成情報を保管します。

## 関連情報

162 ページの『**nimol\_config** コマンド』、165 ページの『**nimol\_install** コマンド』、167 ページの『**nimol\_lslpp** コマンド』。

---

## nimquery コマンド

### 目的

NIM 定義情報を得るためにマシンを照会します。 NIM 環境にクライアント・オブジェクトを作成します。

### 構文

```
nimquery -a host=hostname  
[-a name=client_obj]  
[-d] [-p] [-q] [-v]
```

## 説明

**nimquery** コマンドは、システム情報を得るためにマシンを照会します。この情報は、NIM 環境で新しいクライアント・オブジェクトを定義するために使用されます。システム情報は、NIM Service Handler (nimsh) を使用してマシンから提供されます。

## フラグ

- a 次の attribute=value の組を割り当てます。
- d 新しいクライアント・オブジェクトを定義します (名前属性が必要です)。
- p 精密な印刷形式を使用可能にします。
- q **nimquery** コマンドの属性リストを表示します。
- v コマンド実行時の詳細デバッグ出力を有効にします。

## パラメーター

- host=hostname** 照会するシステムのホスト名を指定します。この属性は必要です。
- name=client\_obj** NIM データベースに新しい定義を作成するときに、クライアントに割り当てる名前を指定します。

## 終了状況

- 0 成功するとゼロを戻します。

## セキュリティ

**nimquery** コマンドを実行するには、root 権限を持っている必要があります。

## 例

1. システム情報を得るためにマシン buckey を照会するには、次のように入力します。  

```
nimquery -a host=buckey
```
2. システム情報を得るためにマシン buckey を照会し、詳細な情報を出力するには、次のように入力します。  

```
nimquery -a host=buckey -p
```
3. NIM オブジェクト名として名前 client6 を使用して、マシン buckey.austin.ibm.com を定義するには、次のように入力します。  

```
nimquery -a name=client6 -a host=buckey -d
```

## ファイル

**/usr/sbin/nimquery** **nimquery** コマンドの位置。

## 関連情報

**nim** コマンド、**nimconfig** コマンド、**nimdef** コマンド。

---

## nis\_cachemgr デーモン

### 目的

NIS+ キャッシュ・マネージャー・デーモンを始動します。

### 構文

```
nis_cachemgr [ -i ] [ -n [ -v ]
```

### 説明

**nis\_cachemgr** デーモンは、NIS+ ディレクトリー・オブジェクトのキャッシュを維持します。このキャッシュには、ネーム・スペース内のさまざまなディレクトリーにサービスを行う NIS+ サーバーに接続するために必要な位置情報が含まれています。ここには、トランスポート・アドレス、サーバーを認証するために必要な情報、ディレクトリー・オブジェクトがキャッシュされる時間の情報を含む「存続時間」フィールドがあります。このキャッシュによって、NIS+ ネーム・スペースを通過するクライアントのパフォーマンスを向上させることができます。**nis\_cachemgr** デーモンは、NIS+ を使用するマシン全部で実行していなければなりません。また **nis\_cachemgr** デーモンは、サービスの対象となる NIS+ 要求に対して実行されている必要があります。

このデーモンによって維持されているキャッシュは、そのマシン上の NIS+ にアクセスするすべてのプロセスによって共有されます。このキャッシュは、すべてのプロセスによってメモリー・マップされたファイル内で維持されます。始動時には、**nis\_cachemgr** デーモンは、コールド・スタート・ファイルからキャッシュを初期化し、キャッシュ・ファイル内に既に存在する期限切れでないエントリーを保持します。したがって、キャッシュはマシン・リブート後もそのまま残ります。

**nis\_cachemgr** デーモンは、通常システム始動スクリプトから始動されます。**nis\_cachemgr** デーモンは、それが実行するホストの NIS+ プリンシパル名のもとで NIS+ 要求を行います。**nis\_cachemgr** デーモンを実行する前に、そのホストのセキュリティー認証を、**nisaddcred** コマンドを使用してそのホストのドメイン内の **cred.org\_dir** テーブルに追加する必要があります。NIS+ サービスがセキュリティー・レベル 2 (**rpc.nisd** コマンドを参照) で実行している場合は、DES タイプの認証が必要です。さらに、**keylogin -r** をそのマシンで実行する必要があります。

**重要:** ホスト・プリンシパルがそのドメイン用の **cred.org\_dir** テーブル内に適切なセキュリティー認証を持たない場合、**-n** 非セキュア・モード・フラグを指定せずにこのデーモンを実行すると、NIS+ 要求を発行するプロセスのパフォーマンスが著しく低下する可能性があります。

### フラグ

- i** **nis\_cachemgr** デーモンが、直前のキャッシュ・ファイルを無視し、コールド・スタート・ファイルからのキャッシュだけを初期化するように強制します。デフォルトにより、キャッシュ・マネージャーは、コールド・スタート・ファイルと古いキャッシュ・ファイルの両方からそれぞれ自身を初期化するため、マシンのリブート後もキャッシュ内のエントリーを維持します。
- n** **nis\_cachemgr** デーモンを無保護モードで実行します。デフォルトにより、そのマシン上での別プロセスの要求時に、共有キャッシュにディレクトリー・オブジェクトを追加する前に、このデーモンは、その要求上の暗号化された署名を調べて、そのディレクトリー・オブジェクトが有効なものであり、許可されたサーバーから送信されたものであるか確かめます。このモードでは、**nis\_cachemgr** デーモンは、このチェックを行わずに共有キャッシュにディレクトリー・オブジェクトを追加します。

- v 詳細モードを設定します。このモードでは、**nis\_cachemgr** デーモンは、エラー・メッセージと警告メッセージだけでなく、追加の状況メッセージもログに記録します。追加のメッセージは、**syslog** を使って優先順位 LOG\_INFO でログを取ります。

## 診断

**nis\_cachemgr** デーモンは、**syslog** を使ってエラー・メッセージと警告メッセージをログに記録します。エラー・メッセージは優先順位 LOG\_ERR で、警告メッセージは優先順位 LOG\_WARNING で、それぞれ DAEMON 機能にログ記録されます。追加の状況メッセージは、**-v** フラグを使って表示できます。

## ファイル

**/var/nis/NIS\_SHARED\_DIRCACHE**  
**/var/nis/NIS\_COLD\_START**  
**/etc/init.d/rpc**

共用キャッシュ・ファイルが入っています。  
コールド・スタート・ファイルが入っています。  
NIS+ の初期化スクリプトが入っています。

## 関連情報

**keylogin** コマンド、**nisaddcred** コマンド、**nisinit** コマンド、**nisshowcache** コマンド。

**rpc.nisd** デーモン。

---

## nisaddcred コマンド

### 目的

NIS+ クレデンシャル情報を作成します。

### 構文

```
nisaddcred [ -p principal ] [ -P nis_principal ] [ -l login_password ] auth_type [ domain_name ]
```

```
nisaddcred -r [ nis_principal ] [ domain_name ]
```

### 説明

**nisaddcred** コマンドは、NIS+ プリンシパルのセキュリティー・クレデンシャルを作成するために使用します。NIS+ クレデンシャルは、2 つの目的のために使用します。第 1 の目的は、さまざまなサービスに認証情報を提供することで、第 2 の目的は、NIS+ プリンシパル名に認証サービス名をマップすることです。

**nisaddcred** コマンドが実行されると、これらのクレデンシャルが作成され、デフォルトの NIS+ ドメイン内にあるテーブル **cred.org\_dir** に格納されます。**domain\_name** を指定すると、エントリーは指定ドメインの **cred.org\_dir** 内に保管されます。指定ドメインは、ユーザーの所属ドメインか、ユーザーがクレデンシャルの作成を認証されていて、その許可を持っているドメイン、すなわちサブドメインのいずれかでなければなりません。通常ユーザーのクレデンシャルは、そのユーザーのパスワードと同じドメイン内に格納されている必要があります。

クレデンシャルの追加は **nisclient** コマンドを使用する方が簡単です。このコマンドは、それ自身で必要な情報を入手するためです。**nispopulate** コマンドは、大量の更新をする場合に使用され、ホストおよびパスワードの NIS+ テーブルにエントリーのクレデンシャルを追加する場合にも使用できます。

NIS+ プリンシパル名は、NIS+ オブジェクトへのアクセス権限を持つクライアントを指定する際に使用されます。さらに、この他にもさまざまなサービスで、これらのプリンシパル名に基づいたアクセス制御を、インプリメントすることができます。

cred.org\_dir テーブルは次のように編成されます。

<b>cname</b>	<b>auth_type</b>	<b>auth_name</b>	<b>public_data</b>	<b>private_data</b>
user1.foo.com.	LOCAL	2990	10,102,44	
user1.foo.com.	DES	unix.2990@foo.com	098...819	3b8...ab2

**cname** カラムには、NIS+ プリンシパル名の規範的な表記が含まれます。規則によって、この名前は、ユーザーのログイン名かマシンのホスト名に後にドット (「.」) を付け、さらに、その後にそのプリンシパルの完全修飾ホスト・ドメインを続けたものになります。ユーザーの場合、ホーム・ドメインはその DES クレデンシャルが保持されているドメインになるように定義されます。ホストの場合、そのホーム・ドメインは、そのホスト上で実行された **domainname** コマンドによって戻されたドメイン名になるように定義されます。

cred.org\_dir テーブル内には 2 つのタイプの *auth\_type* エントリーがあります。認証タイプ LOCAL のものと、認証タイプ DES のものです。 *auth\_type* (コマンド行に大文字か小文字で指定されている) は、local または des でなければなりません。

タイプ LOCAL のエントリーは、NIS+ サービスが、完全修飾 NIS+ プリンシパル名と、cred.org\_dir テーブルを含むドメイン内の UID で識別されたユーザーとの間の対応を決定するために使用します。この対応は、AUTH\_SYS RPC 認証フレーバーを使って NIS+ プリンシパル名に対して行われた要求を関連付けるときに必要です。また、1 つのドメイン内の UID を、ホーム・ドメインがどこか別のロケーションにある、その完全修飾 NIS+ プリンシパル名にマップする場合にも必要です。このようにして、認証フレーバーについてのプリンシパルのクレデンシャルが、そのプリンシパルのホーム・ドメイン (プリンシパル名から抽出された) 内の cred.org\_dir テーブルの中で探索されます。同じ NIS+ プリンシパルが複数のドメイン内で LOCAL クレデンシャル・エントリーを持っている場合があります。ユーザーだけ (マシンでなく) が LOCAL クレデンシャルを持つことができます。NIS+ のユーザーは、そのホーム・ドメイン内では両方のタイプのクレデンシャルを持つことができます。

LOCAL タイプのエントリーに関連している *auth\_name* は、cred.org\_dir テーブルが入っているドメイン内のプリンシパルの場合に有効な UID です。これは、そのプリンシパルのホーム・ドメイン内のものと異なってもかまいません。このタイプの場合 *public\_data* 内に格納されている公開情報には、そのユーザーがメンバーになっているグループの GID のリストが入っています。それらの GID は、このテーブルが常駐するドメインにも適用されます。このタイプに関連した秘密データはありません。UID もプリンシパル名も、1 つの cred.org\_dir テーブルの LOCAL エントリーの中に複数回表示しないようにしてください。

DES *auth\_type* は、セキュア RPC 認証に使用されます。

DES *auth\_type* に関連している認証名は、セキュア RPC ネット名です。セキュア RPC ネット名の形式は `unix.id@domain.com` です。この場合、*domain* は、プリンシパルのドメインと同じでなければなりません。プリンシパルがユーザーの場合、ID は、そのプリンシパルのホーム・ドメイン内のプリンシパルの UID でなければなりません。プリンシパルがホストの場合、ID はそのホストの名前です。セキュア RPC では、有効な UID 0 (root) のもとで実行しているプロセスは、ホスト・プリンシパルによって識別されます。LOCAL とは異なり、NIS+ ネーム・スペース内の 1 つの NIS+ プリンシパルに、複数の DES クレデンシャル・エントリーがあってはなりません。



タイプ DES の認証のエントリー内の公開情報は、そのプリンシパルの公開鍵です。このエントリー内の専用情報は、該当のプリンシパルのネットワーク・パスワードによって暗号化されたそのプリンシパルの秘密鍵です。

NIS+ のユーザー・クライアントは、そのホーム・ドメイン内では両方のタイプのクレデンシャルが必要です。さらに、プリンシパルは、そのプリンシパルが認証された要求を出したいと希望している各ドメインの `cred.org_dir` テーブル内に、LOCAL エントリーを持っていなければなりません。LOCAL エントリーを持たないドメインから要求を出している NIS+ のクライアントは、DES クレデンシャルを獲得できません。セキュリティー・レベル 2 以上で実行している NIS+ サービスは、このようなユーザーを認証されていないものと見なし、アクセス権限を決定する場合に、それらのユーザーに **nobody** という名前を割り当てます。

このコマンドは、認証テーブル内のエントリーを追加または削除する許可を持つ NIS+ プリンシパルによってしか実行できません。

呼び出し側そのものでクレデンシャルが追加されている場合、**nisaddcred** が自動的にその呼び出し側のキー・ログインを実行します。

**nismatch** を使うと、特定のプリンシパルの `cred` エントリーがリストできます。

## フラグ

### **-l** *login\_password*

クレデンシャル・エントリーの秘密鍵を暗号化するには、パスワードとして指定する *login\_password* を使用します。これは、シェルからのパスワードのプロンプトをオーバーライドします。このフラグは、管理スクリプト専用のものです。プロンプトは、**ps** コマンドを使用してコマンド行のパスワードを見る人がいないことを保証し、間違いないことのチェックも行います。

**注:** *login\_password* は必ずしもユーザーのパスワードでなくてもかまいませんが、ユーザーのパスワードである場合はログインが簡単になります。

### **-p** *principal*

その特定のメカニズムの命名規則で定義されたように、プリンシパルの名前を指定します。例えば、LOCAL クレデンシャル名には、UID を指定した文字列を組み込むことによって、このフラグが提供されます。DES クレデンシャルの場合、この名前は、前述のように形式 `unix.id@domain.com` のセキュア RPC ネット名にする必要があります。**-p** フラグを指定しない場合、*auth\_name* フィールドは、現行プロセスの有効 UID と、ローカル・ドメインの名前から作成されます。

### **-P** *nis\_principal*

NIS+ プリンシパル名 *nis\_principal* を使用します。このフラグは、ホーム・ドメインがローカル・マシンのデフォルト・ドメインとは異なるユーザーの、LOCAL クレデンシャルまたは DES クレデンシャルを作成するときに使用するようになっています。**-P** フラグが指定されない場合は、必ず、次のように **nisaddcred** がそのエントリーのプリンシパル名を作成します。**nisaddcred** は、タイプ LOCAL のエントリーを作成しない場合、**nis\_local\_principal** を呼び出します。これは、`cred.org_dir` テーブル内にある現行プロセスの有効 UID に対応する既存の LOCAL エントリーを探し、その新しいエントリーの関連プリンシパル名を使用します。**nisaddcred** は、タイプ LOCAL の認証エントリーを作成するとき、それ自身のプロセスの有効 UID のログイン名を取り、それにドット (「.」) を付加し、さらにその後ローカル・マシンのデフォルト・ドメインを続けることによって、デフォルトの NIS+ プリンシパル名を作成します。呼び出し元がスーパーユーザーの場合、ログイン名の代わりにマシン名が使用されます。



**-r** [ *nis\_principal* ]

プリンシパル *nis\_principal* に関連したクレデンシャルをすべて *cred.org\_dir* テーブルから除去します。このフラグは、システムからクライアントやユーザーを除去するときに使用できます。 *nis\_principal* を指定しない場合、デフォルトによって、現行 *user* のクレデンシャルが除去されます。 *domain\_name* を指定しない場合、デフォルトの NIS+ ドメイン内で操作が実行されます。

## 終了状況

このコマンドは、以下の終了値を戻します。

0	正常終了
1	失敗

## 例

1. UID が 2990 であり、*some.domain.com* NIS+ ドメイン内でプリンシパルな NIS+ ユーザーである *user1* について LOCAL クレデンシャルと DES クレデンシャルを追加するには、次のように入力します。

```
nisaddcred -p 2990 -P user1.some.domain.com. local
```

クレデンシャルは、*domain\_name* がコマンド行の最後のパラメーターとして指定されていない限り、常に **nisaddcred** が実行されるドメイン内の *cred.org\_dir* テーブル内に追加されます。クレデンシャルが、そのクライアントのドメイン・サーバーから追加されている場合、*domain\_name* を指定する必要があります。呼び出し元は、*cred.org\_dir* テーブルにエントリーを作成するための適切な許可を持っている必要があります。

2. 同じユーザーで DES クレデンシャルを追加するには、システム管理者が次のように入力します。

```
nisaddcred -p unix.2990@some.domain.com -P user1.some.domain.com. des
```

DES クレデンシャルは、LOCAL クレデンシャルが追加された後でないと追加できません。セキュア RPC ネット名はドット (「.」) で終了することはありませんが、NIS+ プリンシパル名 (**-P** フラグによって指定) はドットで終了します。このコマンドは、ユーザーと同じドメイン内のマシンから実行してください。

3. 同じドメイン内でマシンの DES クレデンシャルを追加するには、次のように入力します。

```
nisaddcred -p unix.foo@some.domain.com -P foo.some.domain.com. des
```

この場合は LOCAL クレデンシャルは必要ありません。

4. NIS+ ワークステーションのプリンシパル DES クレデンシャルを追加する場合は、次のように入力します。

```
nisaddcred -p unix.host1@sub.some.domain.com \  
-P newhost.sub.some.domain.com. des sub.some.domain.com.
```

このフォーマットは、*sub.some.domain.com* より高いドメイン内にあるサーバーからこのコマンドを実行している場合には特に有用です。ドメイン名の最後のオプションを指定しないと、**nisaddcred** は、*some.domain.com* のデフォルトのドメイン名を使用しようとするために失敗します。

5. root ログイン・パスワードのプロンプトが出されていない場合に DES クレデンシャルを追加するには、次のように入力します。

```
nisaddcred -p unix.2990@some.domain.com -P user1.some.domain.com. -l  
login_password des
```

## 関連コマンド

**chkey** コマンド、 **domainname** コマンド、 **keylogin** コマンド、 **niscat** コマンド、 **nischmod** コマンド、 **nischown** コマンド、 **nisclient** コマンド、 **nismatch** コマンド、 **nispopulate** コマンド、 **ps** コマンド。

---

## nisaddent コマンド

### 目的

対応する **/etc** ファイルまたは NIS マップから NIS+ テーブルを作成します。

### 構文

```
nisaddent [ -D defaults ] [ -P ] [ -a ] [ -r ] [ -v ] [ -t table ] type [ nisdomain ]
```

```
nisaddent [ -D defaults ] [ -P ] [ -a ] [ -p ] [ -r ] [ -m ] [ -v ] -f file [ -t table ] type [ nisdomain ]
```

```
nisaddent [ -D defaults ] [ -P ] [ -a ] [ -r ] [ -m ] [ -v ] [ -t table ] -y ypdomain [ -Y map ] type [ nisdomain ]
```

```
nisaddent -d [ -A ] [ -M ] [ -q ] [ -t table ] type [ nisdomain ]
```

### 説明

**nisaddent** コマンドは、NIS+ テーブル内のエントリーを、それに対応する **/etc** ファイルおよび NIS マップから作成します。この操作は、システムの管理で使用されるそれぞれの標準テーブルごとにカスタマイズされます。 *type* 引数は、処理されるデータのタイプを指定します。このタイプで正しい値は、標準テーブルの場合、**aliases**、**bootparams**、**ethers**、**group**、**hosts**、**netid**、**netmasks**、**networks**、**passwd**、**protocols**、**publickey**、**rpc**、**services**、**shadow** または **timezone** のうちの 1 つで、総称 2 カラム (キー、値) テーブルでは **key-value** です。サイト特定のテーブルの場合 (**key-value** タイプでない)、その管理に **nistbladm** が使用できます。

NIS+ テーブルは、**nistbladm**、**nissetup**、または **nissserver** によって既に作成されているはずですが。

システム・テーブルを転送する場合は、**nisaddent** の代わりに **nispopulate** を使用する方が簡単です。

デフォルトによって、**nisaddent** は標準入力から読み取り、このデータを、コマンド行に指定された *type* に関連した NIS+ テーブルに追加します。 **-t** フラグを使用して、代替 NIS+ テーブルを指定することもできます。タイプ **key-value** の場合、テーブルの指定が必要です。

**注:** *data* タイプは、テーブル名 (**-t**) と異なっていてもかまいません。例えば、自動マウンター・テーブルのテーブル・タイプは **key-value** です。

*shadow* データ・タイプがあっても、対応する *shadow* テーブルはありません。 *shadow* データと *passwd* データは両方とも **passwd** テーブルそのものの中に格納されます。

ファイルは、**-f** フラグを使って処理でき、NIS バージョン 2 (YP) マップは **-y** フラグを使って処理できます。 **-m** フラグは、データを標準入力から読み取る時には使用できません。

*ypdomain* を指定すると、**nisaddent** コマンドは、その入力を、該当の NIS マップ (**mail.aliases**、**bootparams**、**ethers.byaddr**、**group.byname**、**hosts.byaddr**、**netid.byname**、**netmasks.byaddr**、**networks.byname**、**passwd.byname**、**protocols.byname**、**publickey.byname**、**rpc.bynumber**、

**services.byname**、または **timezone.byname**) の **dbm** ファイルから取ります。 **-Y** フラグを指定して、代替の NIS マップを指定することもできます。タイプ **key-value** の場合、マップの指定が必要です。このマップは、ローカル・マシン上の **/var/yp/ypdomain** ディレクトリー内になければなりません。

**注:** *ypdomain* は大文字小文字を区別します。 **ypxfr** コマンドは、NIS マップを獲得するために使用できます。

*nisdomain* を指定すると、**nisaddent** は、その NIS+ ドメイン内の NIS+ テーブルで動作します。指定しない場合、デフォルトのドメインが使用されます。

パフォーマンスに関する場合、テーブルのロードは、**dbm** ファイル (**y**) を使用して行ったときの方が速くできます。

## フラグ

- a** 既存のエントリーを削除せずに、ファイルまたはマップを NIS+ テーブルに追加します。このフラグがデフォルトです。このモードは、削除ではなく、追加と修正だけを伝搬します。
- A** そのテーブル内のデータと、初期テーブルの連結パス内のテーブル内の全データを戻すように指定します。
- d** NIS+ テーブルを、所定の *type* の適切な形式の標準出力にダンプします。タイプ **key-value** のテーブルの場合は、代わりに **niscat** を使用します。クレデンシャル・テーブルをダンプするには、**publickey** と **netid** タイプをダンプします。
- D defaults** この操作中に、異なるデフォルト・セットを使用するように指定します。  
*defaults* 文字列は、コロンで区切られた一連のトークンです。これらのトークンは、汎用オブジェクト属性に使用するデフォルト値を表します。正しいトークンをすべて、以下で紹介します。
- ttl=time** このコマンドによって作成されたオブジェクトのデフォルトの存続時間を設定します。値 *time* は、**nischttl** コマンドによって定義された形式で指定されます。デフォルトは 12 時間です。
- owner=ownername**  
NIS+ プリンシパル *ownername* が、作成されたオブジェクトを所有するように指定します。この値のデフォルトは、そのコマンドを実行しているプリンシパルです。
- group=groupname**  
グループ *groupname* が、作成されたオブジェクトのグループ・オーナーとなるように指定します。デフォルトは **NULL** です。
- access= rights**  
所定のオブジェクトに関して認可される 1 組のアクセス権限を指定します。値 *rights* は、**nischmod** コマンドによって定義された形式で指定されます。デフォルトは **---rmcdr--r--** です。
- f file** *file* を、入力 (標準入力ではなく) のソースとして使用するよう指定します。

<b>-m</b>	ファイルまたはマップを NIS+ テーブルとマージします。これは、変更の数が少ない場合に、ファイルまたは NIS マップによって NIS+ テーブルを最新のものにする、最も効率的な方法です。このフラグは、まだデータベース内にないエントリーを追加し、既に存在しているエントリーが変更されていたらそれを修正し、ソース内にないエントリーを削除します。 <b>-m</b> フラグは、該当のデータベースが大きくて複製されており、ロードされるマップが、いくつかのエントリーにおいてだけ異なっている場合には、必ず使用します。このフラグは、レプリカに送信しなければならない更新メッセージの数を減らします。 <b>-r</b> フラグも参照してください。
<b>-M</b>	ルックアップをマスター・サーバーに送信するように指定します。これによって、最新の情報を表示することが保証されますが、場合によってはマスター・サーバーが使用中であったり、この操作によってマスター・サーバーを使用中にさせたりする、という欠点があります。
<b>-p</b>	ファイルからパスワード情報をロードするときにパスワード・フィールドを処理します。デフォルトによって、パスワード・フィールドは、通常は無効であるために (実際のパスワードはシャドー・ファイル内に表示されます) 無視されます。
<b>-P</b>	初期検索が失敗であった場合、ルックアップをテーブルの連結パスの後に続けるように指定します。
<b>-q</b>	「高速」モードでテーブルをダンプします。テーブルをダンプするためのデフォルト・メソッドでは、各エントリーが個々に処理されます。テーブルによっては (ホストなど)、複数のエントリーを 1 行に結合して、サーバーへ追加の要求を出さなければならないものがあります。「高速」モードでは、テーブルのエントリーがすべて、サーバーへの 1 回の呼び出しで検索されるため、テーブルをより迅速にダンプすることができます。ただし、大きいテーブルの場合、プロセスで仮想メモリーが不足して、そのテーブルがダンプされない可能性があります。
<b>-r</b>	既存の NIS+ テーブル内のファイルまたはマップを、最初に既存エントリーを削除してから、ソース ( <b>/etc</b> ファイルまたは NIS+ マップ) からエントリーを追加することによって置き換えます。このフラグは、 <b>-m</b> フラグと同じ働きがあります。このフラグは、変更が大量にある場合を除き、パフォーマンスへの悪影響を考慮して、できるだけ使用しないようにお勧めします。
<b>-t table</b>	<i>table</i> が、この操作の NIS+ テーブルになるように指定します。これは、デフォルトのドメイン、または <i>domainname</i> が指定されている場合はこの名前と比較した、相対名でなければなりません。
<b>-v</b>	詳細モードを設定します。
<b>-y yppdomain</b>	入力のソースとして、NIS ドメイン <i>yppdomain</i> から、該当の NIS マップの <b>dbm</b> ファイルを使用します。このファイルは、 <b>/var/yp/yppdomain</b> ディレクトリー内のローカル・マシン上にあると考えられています。そのマシンが NIS サーバーでない場合は、 <b>ypxfr</b> コマンドを使用して、該当のマップの <b>dbm</b> ファイルのコピーを入手します。
<b>-Y map</b>	入力のソースとして、 <i>map</i> の <b>dbm</b> ファイルを使用します。

## 環境

### NIS\_DEFAULTS

この変数には、NIS+ 標準デフォルトをオーバーライドするデフォルト文字列が入っています。 **-D** フラグを使用する場合、これらの値は、**NIS\_DEFAULTS** 変数と標準デフォルトを両方ともオーバーライドします。セキュリティ上の事故を防止するため、**NIS\_DEFAULTS** 変数内のアクセス権限は **passwd** テーブルに関しては無視されますが、**-D** フラグによって指定されたアクセス権限は使用されます。

### NIS\_PATH

この変数が設定されていて、*nisdomain* も *table* も完全修飾でない場合、**NIS\_PATH** に指定された各ディレクトリが、そのテーブルが見つかるまで検索されます (**nisdefaults** コマンドを参照)。

## 終了状況

このコマンドは、以下の終了値を戻します。

0	正常終了
1	構文解析以外のエラーが原因で失敗しました。
2	エントリーで構文解析エラーが起きました。構文解析エラーが起こっても終了しません。無効なエントリーがスキップされるだけです。

## 例

1. **/etc/passwd** の内容を **passwd.org\_dir** テーブルに追加するには、次のように入力します。

```
cat /etc/passwd | nisaddent passwd
```

2. シャドー情報を追加するには、次のように入力します。

```
cat /etc/shadow | nisaddent shadow
```

テーブル・タイプは、実際の情報が **passwd** テーブル内に格納されている場合でも、**passwd** でなく **shadow** です。

3. **hosts.org\_dir** テーブルを、**/etc/hosts** の内容と置き換える (詳細モード) には、次のように入力します。

```
nisaddent -rv -f /etc/hosts hosts
```

4. **myppdomain** からの **passwd** マップを、**passwd.org\_dir.nisdomain** テーブルとマージする (詳細モード) には、次のように入力します。

```
nisaddent -mv -y myppdomain passwd nisdomain
```

この例では、**/var/yp/myppdomain** ディレクトリーに **yppasswd** マップが入っているものとしています。

5. **myppdomain** からの **auto.master** マップを **auto\_master.org\_dir** テーブルとマージするには、次のように入力します。

```
nisaddent -m -y myppdomain -Y auto.master -t auto_master.org_dir key-value
```

6. **hosts.org\_dir** テーブルをダンプするには、次のように入力します。

```
nisaddent -d hosts
```

## 関連情報

**niscat** コマンド、 **nischmod** コマンド、 **nisdefaults** コマンド、 **nispopulate** コマンド、 **nisserver** コマンド、 **nissetup** コマンド、 **nistbladm** コマンド、 **passwd** コマンド、 **ypxfr** コマンド。

---

## niscat コマンド

### 目的

NIS+ テーブルの内容を表示します。

### 構文

```
niscat [ -A ] [ -h ] [ -L ] [ -M ] [ -v ] tablename
```

```
niscat [ -A ] [ -L ] [ -M ] [ -P ] -o name
```

### 説明

最初の構文では、**niscat** コマンドは、*tablename* で指定された NIS+ テーブルの内容を表示します。2 番目の構文では、*name* によって指定された NIS+ オブジェクトの内部表現を表示します。

### フラグ

- A** そのテーブル内のデータと、初期テーブルの連結パス内のテーブルの全データを表示します。
- h** テーブルの表示前にヘッダー行を表示します。ヘッダーでは、# 文字と、その後に各カラムの名前が続きます。カラム名は、そのテーブルのセパレーター文字によって区切られます。
- L** リンクの後に続きます。*tablename* か *name* で LINK タイプ・オブジェクトが指名されている場合にこのフラグを指定すると、そのリンクの後に続くことになり、そのリンクで指名されたオブジェクトまたはテーブルが表示されます。
- M** 指名されたデータのマスター・サーバーに要求を送信するように指定します。これによって、最新の情報を表示することが保証されますが、場合によってはマスター・サーバーにかかる負荷が増えたり、更新の場合に NIS+ サーバーが用不可になったり、使用中になったりする可能性が高くなるという欠点があります。
- o** 指名された NIS+ オブジェクトの内部表現を表示します。*name* が索引付きの名前の場合、それぞれの突き合わせエントリー・オブジェクトが表示されます。個々のカラムのアクセス権限などの属性を表示する場合は、このフラグを使用します。
- P** 連結パスの後に続きます。このフラグは、初期検索が失敗であった場合、要求をテーブルの連結パスの後に続けるように指定します。このフラグは、*name* の索引付きの名前と **-o** フラグを使用している場合にのみ有効です。
- v** バイナリー・データを直接表示します。このフラグは、標準出力上のバイナリー・データを含むカラムを表示します。このフラグを使用しない場合、バイナリー・データは文字列 \*BINARY\* として表示されます。

### 環境

**NIS\_PATH** この変数が設定されていて、NIS+ 名が完全修飾でない場合、指定された各ディレクトリーが、該当のオブジェクトが見つかるまで検索されます (**nisdefaults** コマンドを参照)。



## 終了状況

このコマンドは、以下の終了値を戻します。

0	正常終了
1	失敗

## 例

1. ホストのテーブルの内容を表示するには、次のように入力します。

```
niscat -h hosts.org_dir
# cname name addr comment
client1 client1 129.144.201.100 Joe Smith
crunchy crunchy 129.144.201.44 Jane Smith
crunchy softy 129.144.201.44
```

ユーザーが不十分なアクセス権限しか持たない場合、これらのフィールドに文字列 \*NP\* が戻されません。

2. 標準出力上に passwd.org\_dir を表示するには、次のように入力します。

```
niscat passwd.org_dir
```

3. テーブル frodo の内容と、その連結パス内の全テーブルの内容を表示するには、次のように入力します。

```
niscat -A frodo
```

4. テーブル groups.org\_dir 内のエントリーを NIS+ オブジェクトとして表示するには、次のように入力します。

```
niscat -o '[ ]groups.org_dir'
```

大括弧は、単一引用符によってシェルから保護されています。

5. passwd.org\_dir テーブルのテーブル・オブジェクトを表示するには、次のように入力します。

```
niscat -o passwd.org_dir
```

上記の例では、passwd テーブルでなく、passwd テーブル・オブジェクトが表示されます。このテーブル・オブジェクトには、カラムの数、カラム・タイプ、検索可能または検索不能セパレーター、アクセス権限、その他のデフォルトなどの情報が含まれます。

6. org\_dir のディレクトリー・オブジェクト (アクセス権限やレプリカ情報などの情報を含む) を表示するには、次のように入力します。

```
niscat -o org_dir
```

## 関連情報

nistbladm コマンド、nisdefaults コマンド、nismatch コマンド。

---

## nischgrp コマンド

### 目的

NIS+ オブジェクトのグループ・オーナーを変更します。



## 構文

`nischgrp [ -A ] [ -f ] [ -L ] [ -P ] group name`

## 説明

`nischgrp` コマンドは、*name* よって指定された NIS+ オブジェクトまたはエントリーのグループ・オーナーを、指定の NIS+ *group* に変更します。エントリーは、索引付きの名前を使用して指定されます。*group* が完全修飾 NIS+ グループ名でない場合、その名前はディレクトリー検索パスを使って解決されます。追加情報は、`nisdefaults` コマンドを参照してください。

オブジェクトのグループ・オーナーを変更する場合の唯一の制限は、そのオブジェクトに関する修正許可を保有していなければならない、ということです。

このコマンドは、マスター NIS+ サーバーが実行していなければ失敗します。

NIS+ サーバーは、その修正を有効にする前にグループ名の妥当性を検査します。

## フラグ

<b>-A</b>	<i>name</i> によって指定された検索基準と一致する連結パス内の全テーブル内の全エントリーを修正します。このフラグは <b>-P</b> フラグを暗黙指定します。
<b>-f</b>	操作を強制し、それが正常に実行されなければ何の情報も出さずに失敗します。
<b>-L</b>	リンクの後に続き、リンクそのもののグループ・オーナーでなく、リンクされたオブジェクトまたはエントリーのグループ・オーナーを変更します。
<b>-P</b>	指名されたテーブル内の連結パスの後に続きます。このフラグは、 <i>name</i> が索引付きの名前であるか、 <b>-L</b> フラグも指定されていて、指名されたオブジェクトがエントリーを指すリンクであるときに有効です。

## 環境

<b>NIS_PATH</b>	この変数が設定されていて、NIS+ 名が完全修飾でない場合、指定された各ディレクトリーが、該当のオブジェクトが見つかるまで検索されます ( <code>nisdefaults</code> コマンドを参照)。
-----------------	--

## 終了状況

このコマンドは、以下の終了値を戻します。

0	正常終了
1	失敗

## 例

1. オブジェクトのグループ・オーナーを、異なるドメイン内のグループに変更するには、次のように入力します。

```
nischgrp newgroup.remote.domain. object
```

2. オブジェクトのグループ・オーナーを、ローカル・ドメイン内のグループに変更するには、次のように入力します。

```
nischgrp my-buds object
```

3. パスワード・エントリーのグループ・オーナーを変更するには、次のように入力します。

```
nischgrp admins '[uid=99],passwd.org_dir'
```

**admins** は、同じドメイン内の NIS+ グループです。

4. リンクによって指し示されているオブジェクトまたはエントリーのグループ・オーナーを変更するには、次のように入力します。

```
nischgrp -L my-buds linkname
```

5. **hobbies** テーブル内の全エントリーのグループ・オーナーを変更するには、次のように入力します。

```
nischgrp my-buds '[],hobbies'
```

## 関連情報

**nischmod** コマンド、**nischown** コマンド、**nisdefaults** コマンド、**nisgrpadm** コマンド。

---

## nischmod コマンド

### 目的

NIS+ オブジェクトに関するアクセス権限を変更します。

### 構文

```
nischmod [ -A ] [ -f ] [ -L ] [ -P ] mode name...
```

### 説明

**nischmod** コマンドは、*name* によって指定された NIS+ オブジェクトまたはエントリーのアクセス権限 (モード) を *mode* に変更します。エントリーは、索引付きの名前を使用して指定されます。オブジェクトへの修正アクセスのあるプリンシパルだけが、そのモードを変更できます。

*mode* は次の形式になります。

```
rights [,rights]...
```

*rights* の形式は次のとおりです。

```
[ who ] op permission [ op permission ]...
```

*who* は、以下の組み合わせです。

<b>n</b>	Nobody (該当者なし) の許可
<b>o</b>	Owner (オーナー) の許可
<b>g</b>	Group (グループ) の許可
<b>w</b>	World (ワールド) の許可
<b>a</b>	全部または <b>owg</b>

*who* を省略した場合、デフォルトは **a** になります。

*op* は次のいずれかです。

<b>+</b>	<i>permission</i> を認可します。
<b>-</b>	<i>permission</i> を取り消します。

= 許可を明示的に設定します。

*permission* は、以下を任意に組み合わせたものです。

<b>r</b>	読み取り
<b>m</b>	修正
<b>c</b>	作成
<b>d</b>	破棄

## フラグ

- A** *name* によって指定された検索基準と一致する連結パス内の全テーブル内の全エンタリーを修正します。このフラグは **-P** フラグを暗黙指定します。
- f** 操作を強制し、それが正常に実行されなければ何の情報も出さずに失敗します。
- L** リnkの後に続き、リンクそのものの許可でなく、リンクされたオブジェクトまたはエンタリーの許可を変更します。
- P** 指名されたテーブル内の連結パスの後に続きます。このフラグは、*name* が索引付きの名前であるか、**-L** フラグも指定されていて、指名されたオブジェクトがエンタリーを指すリンクであるときにのみ適用できます。

## 環境

**NIS\_PATH** この変数が設定されていて、NIS+ 名が完全修飾でない場合、指定された各ディレクトリーが、該当のオブジェクトが見つかるまで検索されます (**nisdefaults** コマンドを参照)。

## 終了状況

このコマンドは、以下の終了値を戻します。

0	正常終了
1	失敗

## 例

- 全員にオブジェクトへの読み取りアクセス (すなわち、オーナー、グループ、および全部のアクセス) を与えるには、次のように入力します。  
`nischmod a+r object`
- group** および認証されていないクライアント (**nobody**) に対する作成と修正の特権をリジェクトするには、次のように入力します。  
`nischmod gn-cm object`
- オブジェクトに関する複合許可セットを設定するには、次のように入力します。  
`nischmod o=rmcd,g=rm,w=rc,n=r object`
- パスワード・テーブル内のエンタリーの許可を設定して、グループ・オーナーがそれらを修正できるようにするには、次のように入力します。  
`nischmod g+m '[uid=55],passwd.org_dir'`
- リンクされたオブジェクトの許可を変更するには、次のように入力します。

```
nischmod -L w+mr linkname
```

## 関連情報

**chmod** コマンド、**nischgrp** コマンド、**nischown** コマンド、**nisdefaults** コマンド。

---

## nischown コマンド

### 目的

1 つまたは複数の NIS+ オブジェクトまたはエントリーのオーナーを変更します。

### 構文

```
nischown [ -A ] [ -f ] [ -L ] [ -P ] owner name...
```

### 説明

**nischown** コマンドは、*name* によって指定された NIS+ オブジェクトまたはエントリーのオーナーを *owner* に変更します。エントリーは、索引付きの名前を使用して指定されます。*owner* が完全修飾 NIS+ プリンシパル名 (**nisaddcred** コマンドを参照) でない場合、デフォルト・ドメイン (**nisdefaults** コマンドを参照) がそれに付加されます。

オブジェクトのオーナーを変更する場合の唯一の制限は、そのオブジェクトに関する修正許可を保有していなければならないということです。

**注:** オブジェクトの現在のオーナーが所有権を変更する場合、そのオーナーは、新しいオブジェクトに対する修正アクセスを持たない場合、所有権を再獲得できません。

このコマンドは、マスター NIS+ サーバーが実行していないければ失敗します。

NIS+ サーバーは、修正を行う前にその名前の妥当性を検査します。

### フラグ

- A** *name* によって指定された検索基準と一致する連結パス内の全テーブル内の全エントリーを修正します。このフラグは **-P** フラグを暗黙指定します。
- f** 操作を強制し、それが正常に実行されなければ何の情報も出さずに失敗します。
- L** リnkの後に続き、リンクそのもののオーナーでなく、リンクされたオブジェクトまたはエントリーのオーナーを変更します。
- P** 指名されたテーブル内の連結パスの後に続きます。このフラグは、*name* が索引付きの名前であるか、**-L** フラグも指定されていて、指名されたオブジェクトがエントリーを指すリンクであるときにのみ意味があります。

### 環境

**NIS\_PATH** この変数が設定されていて、NIS+ 名が完全修飾でない場合、指定された各ディレクトリーが、該当のオブジェクトが見つかるまで検索されます (**nisdefaults** コマンドを参照)。

## 終了状況

このコマンドは、以下の終了値を戻します。

0	正常終了
1	失敗

## 例

1. オブジェクトのオーナーを、異なるドメイン内のプリンシパルに変更するには、次のように入力します。

```
nischown bob.remote.domain. object
```

2. オブジェクトのオーナーを、ローカル・ドメイン内のプリンシパルに変更するには、次のように入力します。

```
nischown skippy object
```

3. passwd テーブル内のエントリーのオーナーを変更するには、次のように入力します。

```
nischown bob.remote.domain. '[uid=99],passwd.org_dir'
```

4. リンクによって指し示されているオブジェクトまたはエントリーを変更するには、次のように入力します。

```
nischown -L skippy linkname
```

## 関連情報

**nisaddcred** コマンド、**nischgrp** コマンド、**nischttl** コマンド、**nischmod** コマンド、**nisdefaults** コマンド。

---

## nischttl コマンド

### 目的

**nischttl** コマンドは、ネーム・スペース内のオブジェクトまたはエントリーの *time-to-live* の値を変更します。

### 構文

#### オブジェクトの *time-to-live* の値を変更する

```
nischttl [-A] [-L] [-P] [time-to-live] [object-name]
```

#### エントリーの *time-to-live* の値を変更する

```
nischttl [ time-to-live ] [ column=value,... ] [ table-name ] [-A] [-L] [-P]
```

注: *time-to-live* が次のように表される場合、それぞれ説明のようになります。

- **秒数**。文字のない数字は、秒数と解釈されます。そのため、TTL の **1234** は 1234 秒と解釈されます。文字 **s** の前にある数字も、秒数と解釈されます。したがって、TTL の **987s** は 987 秒と解釈されます。日数、時間数、分数と組み合わせて秒数を指定する場合、文字 **s** を使用して、秒の値を識別する必要があります。
- **分数**。文字 **m** の前にある数字は、分数と解釈されます。したがって、TTL の **90m** は 90 分と解釈されます。

- **時間数**。文字 **h** の前にある数字は、時間数と解釈されます。したがって、TTL の **9h** は 9 時間と解釈されます。
- **日数**。文字 **d** の前にある数字は、日数と解釈されます。したがって、TTL の **7d** は 7 日と解釈されます。

注: これらの値は組み合わせて使用されます。例えば、TTL 値 **4d3h2m1s** は、4 日、3 時間、2 分、1 秒の *time-to-live* を指定します。

## 説明

この *time-to-live* の値は、キャッシュ・マネージャーが、キャッシュ・エントリーの満了の時期を判断するために使用します。 *time-to-live* は、合計秒数でも、日数、時間数、分数、および秒数の組み合わせでも指定できます。オブジェクトまたはエントリーに割り当てる *time-to-live* の値は、オブジェクトの安定度に依存します。オブジェクトが頻繁に変更される可能性が大きい場合、*time-to-live* の値を小さく指定してください。オブジェクトが安定していれば、大きい値を指定してください。大きい *time-to-live* の値は 1 週間、小さい値は 1 分より小さいというように指定できます。パスワード・エントリーには、1 日 1 回のパスワード変更に対応できるように、約 12 時間の *time-to-live* の値を指定する必要があります。あまり変更されないテーブルのエントリー (例えば、RPC テーブル内のエントリー) の場合は、この値に数週間を指定することもできます。

### 注

1. オブジェクトの *time-to-live* を変更するには、そのオブジェクトに対する修正権限がなければなりません。テーブル・エントリーの *time-to-live* を変更するには、修正したいテーブル、エントリー、またはカラムに対する修正権限がなければなりません。
2. オブジェクトまたはテーブル・エントリーの現行 *time-to-live* の値を表示するには、**nisdefaults -t** コマンド (NIS+ アクセス権限の管理の説明を参照) を使用します。

## フラグ

- A ユーザーが指定した **column=value** の指定と一致するエントリー全部に、この変更を適用します。
- L リンクの後に付けられ、リンクそのものでなく、リンクされたオブジェクトまたはエントリーに変更を適用します。
- P 条件を満たすエントリーが 1 つ現れるまで、パスの後に付けられます。

## 例

### オブジェクトの *time-to-live* を変更する

1. オブジェクトの *time-to-live* を変更するには、*time-to-live* の値と **object-name** を指定して **nischt1** コマンドを入力します。 **-L** コマンドを追加すると、リンクされたオブジェクトへの変更を拡張できます。

```
nischt1 -L time-to-live object-name
```

2. 秒数を入力することにより、*time-to-live* を秒単位で指定できます。また、接尾部 **s**、**m**、**h**、および **d** を使用して、秒数、分数、日数、時間数を示すことで、日数、時間数、分数、および秒数を組み合わせて指定することもできます。以下に例を示します。

86400 秒の TTL

```
client% nischt1 86400 sales.wiz.com.
```

24 時間の TTL

```
client% nischt1 24h sales.wiz.com.
```

2 日、1 時間、1 分、1 秒の TTL

```
client% nischt1 2d1h1m1s sales.wiz.com.
```

- 最初の 2 つのコマンドは、sales.wiz.com. ディレクトリーの *time-to-live* を 86,400 秒または 24 時間に変更します。3 番目のコマンドは、ホスト・テーブル内の全エントリーの *time-to-live* を、2 日、1 時間、1 分、1 秒に変更します。

## テーブル・エントリーの *time-to-live* を変更する

- エントリーの *time-to-live* を変更するには、索引付きエントリー形式を使用します。**-A**、**-L**、**-P** のうちのどのオプションでも使用できます。

```
nischttl [-ALP] time-to-live [column=value,...],  
table-name
```

- 次の例は上記の例と似ていますが、オブジェクトでなくテーブル・エントリーの値を変更します。

```
client% nischttl 86400 '[uid=99],passwd.org_dir.wiz.com.'  
client% nischttl 24h `[uid=99],passwd.org_dir.wiz.com.'  
client% nischttl 2d1h1m1s `[name=fred],hosts.org_dir.wiz.com'
```

注 C シェル・ユーザーは、引用符を使用して、シェルが大括弧 (I) をメタキャラクターと解釈しないようにする必要があります。

## 関連情報

**defaults** コマンド。

---

## nisclient コマンド

### 目的

NIS+ プリンシパルのための NIS+ クレデンシャルを初期化します。

### 構文

NIS+ プリンシパルの DES クレデンシャルを追加する

```
nisclient -c [-x ] [-o ] [-v ] [-l network_password ] [-d NIS+_domain ] client_name...
```

NIS+ クライアント・マシンを初期化する

```
nisclient -i [-x ] [-v ] -h NIS+_server_host [ -a NIS+_server_addr ] [-d NIS+_domain ] [ -S 0 | 2 ]
```

NIS+ ユーザーを初期化する

```
nisclient -u [-x ] [-v ]
```

ネットワーク・サービス環境を復元する

```
nisclient -r [-x ]
```

### 説明

**nisclient** コマンドは、次を行う場合に使用できます。

- ホストおよびユーザーの NIS+ クレデンシャルを作成する
- NIS+ ホストおよびユーザーを初期化する
- ネットワーク・サービス環境を復元する



NIS+ クレデンシャルは、NIS+ クライアントの認証情報を NIS+ サービスに提供するために使用します。

最初の構文 (**-c**) は、ホストまたはユーザーの個々の NIS+ クレデンシャルを作成するために使用します。新しいクレデンシャルを作成するドメイン内では、必ず NIS+ プリンシパルとしてログインしてください。また、ローカル・クレデンシャル・テーブルへの書き込み許可も必要です。 *client\_name* 引数は、NIS+ ドメイン内では、有効ホスト名またはユーザー名であればどれでも受け入れます (例えば、*client\_name* は必ずホストまたはパスワード・テーブル内になければなりません)。 **nisclient** コマンドは、各 *client\_name* を、**host** と **passwd** の両方のテーブルに照らし合わせて検査し、ホストまたはユーザーの適切な NIS+ クレデンシャルを追加します。

注: ローカル・ドメインの外側で NIS+ クレデンシャルを作成している場合、そのホストまたはユーザーは、ローカルとリモートの両方のドメイン内の **host** テーブルまたは **passwd** テーブル内になければなりません。

デフォルトでは、**nisclient** は、指定されたホストおよびユーザーのクレデンシャル・テーブル内にある既存エントリは上書きしません。上書きをする場合は、**-o** フラグを使用します。クレデンシャルが作成された後、**nisclient** は、ホストまたはユーザーを初期化するためにクライアント・マシンで実行しなければならないコマンドを出力します。**-c** フラグには、クライアントのネットワーク・パスワード (そのクライアントの秘密鍵を暗号化するために使用します) が必要です。ネットワーク・パスワードは、**-l** フラグを付けてコマンド行で指定することもできますが、スクリプトがそれを求めるプロンプトを出す場合もあります。このネットワーク・パスワードは、**nispaswd** コマンドか **chkey** コマンドを使って後で変更することもできます。

**-c** フラグは、**passwd** テーブルと **hosts** テーブル内で定義されている全ユーザーおよびホストの NIS+ クレデンシャルを作成する場合に使用するものではありません。全ユーザーおよびホストのクレデンシャルを定義するには、**nispopulate** コマンドを使用します。

2 番目の構文 (**-i**) は、NIS+ クライアント・マシンを初期化する場合に使用します。**-i** フラグを使用すると、マシンを変換して NIS+ を使用したり、マシンのドメイン名を変更できます。NIS+ クライアントになるマシンでは、スーパーユーザーとしてログインする必要があります。管理者が、既に **nisclient -c** または **nispopulate -C** コマンドを使用して、このホストの NIS+ クレデンシャルを作成しているはずで、管理者が作成したネットワーク・パスワードが必要になります。**nisclient** コマンドは、秘密鍵を暗号化解除するためにネットワーク・パスワードを入力するプロンプトを出してから、秘密鍵/公開鍵の新規セットを生成するために、このマシンの **root** ログイン・パスワードを入力するプロンプトを出します。NIS+ クレデンシャルが、**nisclient -c** を使って管理者によって作成されたのであれば、単に **nisclient** スクリプトによって出力された初期化コマンドを使用すれば (そのコマンドを手で入力する代わりに)、このホストを初期化することができます。

認証されていない NIS+ クライアント・マシンを初期化するには、**-S 0** と一緒に **-i** フラグを使用します。これらのフラグを使用すると、**nisclient -i** フラグはパスワードを求めてきません。

クライアントの初期化プロセス実行中、修正されているファイルは *files.no\_nisplus* としてバックアップされます。通常、クライアントの初期化中に修正されるファイルは、**/etc/defaultdomain**、**/etc/nsswitch.conf** および **/etc/inet/hosts** で、さらにファイル **/var/nis/NIS\_COLD\_START** があればこのファイルも該当します。

注: バックアップ・ファイルが既にある場合、ファイルは保存されません。

**-i** フラグは、DNS を使用してホスト名を解決するために、NIS+ クライアントをセットアップしません。DNS のセットアップについては、DNS の資料を参照してください。(resolv.conf ファイル形式に関する説明を参照してください。)

NIS+ クライアントとしてインストールされた NIS+ root マスター・サーバーまたはマシンの初期化は必要ありません。

3 番目の構文 (**-u**) は、NIS+ ユーザーを初期化するために使用します。必ず、NIS+ クレデンシャルが作成されたドメインの中の NIS+ クライアント・マシンにユーザーとしてログインしている必要があります。管理者が、既に **nisclient** コマンドまたは **nispopulate** コマンドを使って、該当のユーザー名の NIS+ クレデンシャルを作成しているはずですが、そのユーザー名の NIS+ クレデンシャルの作成には、管理者が使用したネットワーク・パスワードが必要になります。**nisclient** コマンドは、秘密鍵を暗号化解除するためにこのネットワーク・パスワードを入力するプロンプトを出してから、秘密鍵/公開鍵の新規セットを生成するために、ログイン・パスワードを入力するプロンプトを出します。

4 番目の構文 (**-r**) は、**nisclient -i** が実行される前に使用していたネットワーク・サービス環境を復元するために使用します。復元されるマシンでは、スーパーユーザーとしてログインする必要があります。復元は、そのマシンが **nisclient -i** を使って初期化されている場合にしか実行できません。これは、復元では **-i** フラグによって作成されたバックアップ・ファイルが使用されるためです。

マシンを初期化した後、またはネットワーク・サービスを復元した後はリブートします。

## フラグ

<b>-a</b> <i>NIS+_server_addr</i>	NIS+ サーバーの IP アドレスを指定します。このフラグは、 <b>-i</b> フラグと一緒にしか使用されません。
<b>-c</b>	NIS+ プリンシパルの DES クレデンシャルを追加します。
<b>-d</b> <i>NIS+_domain</i>	<b>-c</b> フラグと一緒に使用するとき、クレデンシャルを作成する必要がある NIS+ ドメインを指定します。このフラグは、 <b>-i</b> フラグと一緒に使用するとき、新しい NIS+ ドメインの名前を指定します。デフォルトは、現在のドメイン名です。
<b>-h</b> <i>NIS+_server_host</i>	NIS+ サーバーのホスト名を指定します。このフラグは、 <b>-i</b> フラグと一緒にの場合にしか使用されません。
<b>-i</b>	NIS+ クライアント・マシンを初期化します。
<b>-l</b> <i>network_password</i>	クライアントのネットワーク・パスワードを指定します。このフラグは、 <b>-c</b> フラグと一緒にの場合にしか使用されません。このフラグを指定しない場合、スクリプトがネットワーク・パスワードを入力するプロンプトを出します。
<b>-o</b>	既存のクレデンシャル・エントリーを上書きします。デフォルトは、上書きをしないことです。このフラグは、 <b>-c</b> フラグと一緒にの場合にしか使用されません。
<b>-r</b>	ネットワーク・サービス環境を復元します。
<b>-S</b> 0   2	NIS+ クライアントの認証レベルを指定します。レベル 0 は、認証されていないクライアントのレベルで、レベル 2 は認証された (DES) クライアントのレベルです。デフォルトは、レベル 2 認証を指定してセットアップすることです。このフラグは、 <b>-i</b> フラグと一緒にの場合にしか使用されません。 <b>nisclient</b> コマンドは、 <b>-c</b> と <b>-u</b> のどちらのフラグの場合も、レベル 2 認証 (DES) を使用します。レベル 0 認証の場合は、 <b>-u</b> および <b>-c</b> を指定した <b>nisclient</b> を実行する必要はありません。
<b>-u</b>	NIS+ ユーザーを初期化します。
<b>-v</b>	詳細モードでスクリプトを実行します。
<b>-x</b>	エコー・モードをオンにします。スクリプトは、実行しようとしていたコマンドだけを出力します。実際には、それらのコマンドは実行されないので注意してください。デフォルトはオフです。

## 例

1. ローカル・ドメイン内のホスト *dilbert* とユーザー *fred* の DES クレデンシャルを追加するには、次のように入力します。

```
nisclient -c dilbert fred
```

2. ドメイン *xyz.ibm.com.* 内のホスト *dilbert* とユーザー *fred* の DES クレデンシャルを追加するには、次のように入力します。

```
nisclient -c -d xyz.ibm.com. dilbert fred
```

3. ドメイン *xyz.ibm.com.* 内でホスト *dilbert* を NIS+ クライアントとして初期化するには (ただし、*nisplus\_server* はドメイン *xyz.ibm.com.* のサーバー)、次のように入力します。

```
nisclient -i -h nisplus_server -d xyz.ibm.com.
```

スクリプトは、サーバーが **/etc/hosts** ファイル内で見つからない場合、*nisplus\_server* の IP アドレスを入力するプロンプトを出します。 **-d** フラグは、現在のドメイン名が新しいドメイン名と異なる場合にのみ必要です。

4. ドメイン *xyz.ibm.com.* 内でホスト *dilbert* を認証されていない NIS+ クライアントとして初期化するには (ただし、*nisplus\_server* はドメイン *xyz.ibm.com.* のサーバー)、次のように入力します。

```
nisclient -i -S 0 -h nisplus_server -d xyz.ibm.com. -a 129.140.44.1
```

5. ユーザー *fred* を NIS+ プリンシパルとして初期化するには、次のように入力して、NIS+ クライアント・マシン上でユーザー *fred* としてログインします。

```
nisclient -u
```

## ファイル

**/var/nis/NIS\_COLD\_START**

このファイルには、サーバー、そのトランスポート・アドレス、およびマシンのデフォルト・ドメインでサービスを行うセキュア RPC 公開鍵のリストが入っています。

**/etc/defaultdomain**

システム・デフォルト・ドメイン名

**/etc/nsswitch.conf**

ネーム・サービス・スイッチの構成ファイル

**/etc/inet/hosts**

ローカル・ホスト名データベース

## 関連情報

**chkey** コマンド、**keylogin** コマンド、**keyserv** コマンド、**nisaddcred** コマンド、**nisinit** コマンド、**nispopulate** コマンド。

**resolv.conf** ファイル形式。

---

## nisdefaults コマンド

### 目的

ネーム・スペースで現在アクティブな 7 つのデフォルト値を表示します。

### 構文

```
nisdefaults [ -d domain ] [ -g group ] [ -h host ] [ -p principal ] [ -r access_rights ] [ -s search_path ] [ -t time_to_live ] [ -a all(terse) ] [ -v verbose ]
```

## 説明

**nisdefaults** コマンドは、ネーム・スペースで現在アクティブな 7 つのデフォルト値を表示します。NIS+ デフォルトを表示する場合、デフォルト値は次のどちらかです。

- NIS+ ソフトウェアによって提供された事前設定値
- **NIS\_DEFAULTS** 環境変数に指定されたデフォルト (**NIS\_DEFAULTS** 値を設定している場合)

このマシン上で作成されたオブジェクトはすべて、オブジェクトを作成するために使用するコマンドの **-D** フラグによって変更されていない限り、自動的に上記のデフォルト値を獲得します。

## デフォルトのセキュリティ値を設定する

このセクションでは、**nisdefaults** コマンド、**NIS\_DEFAULTS** 環境変数、および **-D** フラグに関連したタスクを実行する方法について説明します。**NIS\_DEFAULTS** 環境変数は、次のデフォルト値を指定します。

- オーナー
- グループ
- アクセス権限
- 存続時間

**NIS\_DEFAULTS** 環境変数で設定された値は、そのシェルを使って作成されたすべての NIS+ オブジェクトに適用されます (ただし、オブジェクトを作成するコマンドで **-D** フラグを使って上書きされている場合は除きます)。

**NIS\_DEFAULTS** 環境変数によって指定されたデフォルト値 (オーナー、グループ、アクセス権限、存続時間) が指定可能なデフォルト値です。**NIS\_DEFAULTS** の値を設定すると、そのシェルから作成されたオブジェクトはすべてこれらのデフォルトを得ます。ただし、コマンドを起動するときに **-D** フラグを使用してデフォルト値を上書きした場合は除きます。

## **NIS\_DEFAULTS** の値を表示する

次の例に示すように、**echo** コマンドを使って、環境変数の設定を確認できます。

```
client% echo $NIS_DEFAULTS
owner=butler:group=gamblers:access=o+rmcd
```

また、**nisdefaults** コマンドを使用すると、ネーム・スペース内でアクティブな NIS+ デフォルトの一般的なリストを表示することもできます。

## デフォルトを変更する

**NIS\_DEFAULTS** 環境変数の値を変更すれば、デフォルトのアクセス権限、オーナー、およびグループが変更できます。そのシェル用に適切な環境コマンド (**csh** または **\$NIS\_DEFAULTS=** の場合は **setenv**、**sh** と **ksh** の場合は **export**) に次の引数を指定して使用します。

- **access=right**. **right** は、コマンドのアクセス権限の指定で説明するフォーマットを使用するアクセス権限です。
- **owner=name**. **name** はオーナーのユーザー名です。
- **group=group**. **group** はデフォルト・グループの名前です。

2 つ以上の引数を、コロンで区切って 1 行に結合させることができます。

```
owner=principal-name:group=group-name
```

## デフォルトの変更 — 例

タスク	例
このコマンドは、デフォルトのアクセス権限としてオーナーの読み取りアクセスを認可します。	<code>client%setenv NIS_DEFAULTS access=o+r</code>
このコマンドは、デフォルトのオーナーを、ホーム・ドメインが <code>Wiz.com.</code> であるユーザー <code>abe</code> となるように設定します。	<code>client%setenv NIS_DEFAULTS owner=abe.wiz.com.</code>
このコマンドは、最初の 2 つの例を 1 つのコード行に結合します。	<code>client%setenv NIS_DEFAULTS access=o+r:owner=abe.wiz.com.</code>

デフォルトが変更されたシェルから作成されたオブジェクトおよびエントリはすべて、指定された新規の値を持つようになります。テーブルのカラムまたはエントリにはデフォルトの設定値を指定できません。カラムとエントリは、単純にテーブルのデフォルトを受け継ぎます。

## NIS\_DEFAULTS の値をリセットする

NIS\_DEFAULTS 変数は、そのシェル用に適切なフォーマットを使用し、引数を使用せずに変数の名前を入力することによって、元の値にリセットすることができます。

### C シェルの場合:

```
client# unsetenv NIS_DEFAULTS
```

### Bourne または Korn シェルの場合:

```
client$ NIS_DEFAULTS=; export NIS_DEFAULTS
```

## フラグ

- d domain** コマンドが入力されたワークステーションのホーム・ドメインを表示します。 `/etc/defaultdomain` 環境変数の値を表示します。
  - g group** このシェルから作成された次のオブジェクトに割り当てられるグループを表示します。  
**NIS\_GROUP** 環境変数の値を表示します。
  - h host** ワークステーションのホスト名を表示します。 `uname -n` 環境変数の値を表示します。
  - p principal** `nisdefaults` コマンドを入力した NIS+ プリンシパルの完全修飾ユーザー名またはホスト名を表示します。 `gethostbyname()` 環境変数の値を表示します。
  - r access\_rights** このシェルから作成された次のオブジェクトまたはエントリに割り当てられるアクセス権限を表示します。フォーマット: `--rmcdr--r--`。 **NIS\_DEFAULTS** 環境変数の値を表示します。
  - s search\_path** 検索パスの構文を表示します。検索パスは、NIS+ が情報を探すときに検索するドメインを示します。 **NIS\_PATH** 環境変数の値が設定されていれば、その値を表示します。 **NIS\_PATH** 環境変数の値を表示します。
  - t time\_to\_live** このシェルから作成された次のオブジェクトに割り当てられる存続時間を表示します。デフォルトは 12 時間です。 **NIS\_DEFAULTS** 環境変数の値を表示します。
  - a all (terse)** 簡略形式の 7 つのデフォルトすべてを表示します。 環境変数の値を表示します。
  - v verbose** 詳細モードで指定された値を表示します。 環境変数の値を表示します。
- 注: これらのオプションを使用して、デフォルト値全部と、それらのサブセットをすべて表示できます。

## 例

1. 詳細形式ですべての値を表示するには、引数を指定せずに `nisdefaults` コマンドを入力します。

```
master% nisdefaults  
Principal Name : topadmin.wiz.com.  
Domain Name   : Wiz.com.
```



```
Host Name      : rootmaster.wiz.com.  
Group Name     : salesboss  
Access Rights  : ----rmcdr---r---  
Time to live   : 12:00:00:00:00  
Search Path    : Wiz.com.
```

2. 簡略形式ですべての値を表示するには、**-a** オプションを追加します。
3. 値のサブセットを表示するには、該当のオプションを使用します。値は簡略モードで表示されます。例えば、アクセス権限と検索パスのデフォルトを表示するには、次のように入力します。

```
rootmaster% nisdefaults -rs  
----rmcdr---r---  
Wiz.com.
```

4. 詳細モードで値のサブセットを表示するには、**-v** フラグを追加します。

---

## niserror コマンド

### 目的

NIS+ エラー・メッセージを表示します。

### 構文

```
niserror error-num
```

### 説明

**niserror** コマンドは、標準出力上の状況値 *error-num* に関連した NIS+ を出力します。これは、シェル・スクリプトが、戻された NIS+ エラー番号をテキスト・メッセージに変換するために使用します。

### 例

エラー番号 20 に関連したエラーを出力するには、次のように入力します。

```
niserror 20  
Not Found, no such name
```

---

## nisgrep コマンド

### 目的

NIS+ テーブルを検索するためのユーティリティ。

### 構文

```
nisgrep [ -A ] [ -c ] [ -h ] [ -M ] [ -o ] [ -P ] [ -s [sep] ] [ -v ]
```

### 説明

**nisgrep** コマンドは、NIS+ テーブルを検索するために使用できます。コマンド **nisgrep** は、単純なテキスト一致ではなく、検索基準における正規表現 **keypat** を受け入れる能力において、**nismatch** コマンドと異なっています。

**nisgrep** はコールバック機能を使用するので、テーブル作成時に特別に検索可能にされたカラムだけを探索するには制限されていません。これによって、**nismatch** に比べて柔軟性はありますが、速度は遅くなります。

**nismatch** ではサーバーが検索を行います。それに対し、**nisgrep** ではサーバーがすべての読み取り可能エントリーを戻してから、クライアントがパターン・マッチングを行います。

どちらのコマンドも、パラメーター **tablename** は、検索されるテーブルの NIS+ 名です。カラム名を指定せずに 1 つのキーまたはキー・パターンだけを指定すると、そのキーまたはキー・パターンは最初のカラムの検索時に適用されます。特定の名前が付いたカラムは、**colname=key** 構文を使用して検索できません。複数のカラムを検索するときは、すべてのカラムで一致するエントリーだけが戻されます。これは、論理結合演算と同じことです。

**nismatch** は検索基準の追加形式 **indexedname** を受け入れます。これはその形式の NIS+ 索引付きの名前です。

```
colname=value, . . . ],tablename
```

## フラグ

- A** すべてのデータ。そのテーブル内のデータと、初期テーブルの連結パス内にあるテーブルの全データを戻します。
- c** 検索基準と一致するエントリー数のカウントだけを出力します。
- h** テーブルのカラムの名前が入っている突き合わせエントリーの前に、ヘッダー行を表示します。
- M** マスター・サーバーのみ。指名されたデータのマスター・サーバーにルックアップを送信します。これによって、最新の情報を表示することが保証されますが、場合によってはマスター・サーバーが使用中であるという欠点があります。
- o** 突き合わせ NIS+ オブジェクトの内部表現を表示します。
- P** 連結パスの後に続きます。初期検索が失敗であった場合、ルックアップをテーブルの連結パスの後に続けるように指定します。
- s sep** このオプションは、テーブルのカラムを区切るために使用する文字を指定します。文字を指定しないと、そのテーブルのデフォルト・セパレーターが使用されます。
- v** 詳細。突き合わせエントリーを表示するときに、バイナリー・データの出力を抑止しません。このオプションを使用しない場合、バイナリー・データは文字列 \* **BINARY** \* として表示されます。

## リターン値

- 0** いくつかのエントリーが正常に突き合わせられました。
- 1** テーブルの検索が正常に行われ、一致は見つかりませんでした。
- 2** エラー状態が起きました。エラー・メッセージも出力されます。

## 例

この例では、**zotz.com.** ドメインの **org\_dir** サブディレクトリー内のテーブル **passwd** を検索します。ユーザー名が **skippy** というエントリーを戻します。この例では、すべての作業がサーバー上で行われます。

```
example% nismatch name=skippy passwd.org_dir.zotz.com.
```

この例は上記の例と類似していますが、**nisgrep** を使用して、テーブル **passwd** 内で **ksh** または **cs**h を使用しているすべてのユーザーを見つけるという点が異なります。

```
example% nisgrep 'shell=[ck]sh' passwd.org_dir.zotz.com.
```

**NIS\_PATH**。この変数が設定されていて、NIS+ テーブル名が完全修飾でない場合、指定された各ディレクトリーは、そのテーブルが見つかるまで検索されます (**nisdefaults** を参照)。



## 関連情報

`niscat` コマンド、`nisdefaults` コマンド、`nisls` コマンド、および `nistbladm` コマンド。

---

## nisgrpadm コマンド

### 目的

NIS+ グループに関する各種の管理操作を作成、削除、および実行します。

注: `nisgrpadm` を使用するには、その操作用に適切なアクセス権限がなければなりません。

### 構文

#### グループを作成または削除する、またはメンバーをリストする

```
nisgrpadm [ -c group_name.domain_name ] [ [ -d ] [ -l group_name ] ]
```

#### メンバーを追加または除去する、またはメンバーがグループに所属しているか判別する

```
nisgrpadm [ [ -a ] [ -r ] [ -t ] group_name ]
```

注: メンバーでは、6 つのメンバーシップをどのように組み合わせることもできます。

### 説明

`nisgrpadm` コマンドには 2 つの主な形式があります。1 つはグループに働くもので、1 つはグループ・メンバーに働くものです。

作成 (`-c`) 以外のすべての操作は、部分的に修飾された `group-names` を受け入れます。ただし、`-c` フラグの場合でも、`nisgrpadm` は、`group-name` 引数内での `groups_dir` の使用は受け入れません。

### フラグ

#### グループを作成または削除する、またはメンバーをリストする

<code>-c group_name.domain_name</code>	NIS+ グループを作成します。そのグループのドメインの <code>groups_dir</code> ディレクトリーに対する作成権がなければなりません。
<code>-d group_name</code>	NIS+ グループを削除します。そのグループのドメイン内 <code>groups_dir</code> ディレクトリーに対する削除権がなければなりません。
<code>-l group_name</code>	NIS+ グループのメンバーをリストします。そのグループ・オブジェクトに対する読み取り権がなければなりません。

#### メンバーを追加または除去する、またはメンバーがグループに所属しているか判別する

<code>-a group_name</code>	NIS+ グループにメンバーを追加します。そのグループ・オブジェクトに対する修正権がなければなりません。
<code>-r group_name</code>	NIS+ グループからメンバーを除去します。そのグループ・オブジェクトに対する修正権がなければなりません。
<code>-t group_name</code>	NIS+ プリンシパルが特定の NIS+ グループのメンバーであるかどうかを確認します。そのグループ・オブジェクトに対する読み取りアクセスがなければなりません。

## 関連情報

`nisdefaults` コマンド。

---

## nisinit コマンド

### 目的

ワークステーションを NIS+ クライアントとして初期化します。

### 構文

#### クライアントを初期化する

```
nisinit [ -c [ -k key_domain ] [ -C coldstart | -H host ] -B ]]
```

#### ルート・マスター・サーバーを初期化する

```
nisinit -r
```

#### 親サーバーを初期化する

```
[ -p Y|D|N parent_domain_host... ]
```

### 説明

`nisinit` コマンドは、NIS+ クライアントとしてワークステーションを初期化します。 `rpc.nisd` コマンドの場合と同様に、`nisinit` コマンドを使用するためにアクセス権限は必要ありませんが、その前提条件と関連タスクについては知っておく必要があります。

### フラグ

**-c** NIS+ クライアントとしてマシンを初期化します。初期化オプションには、`coldstart` による初期化、`hostname` による初期化、ブロードキャストによる初期化の 3 つがあります。最も安全なメカニズムは、トラステッド `coldstart` ファイルからの初期化です。2 番目のオプションは、トラステッド・ホストとして指定した `hostname` を使用する場合の初期化です。3 番目のメソッドは、ブロードキャストによる初期化で、これは最も安全性の低いメソッドです。

#### **-C**`coldstart`

NIS+ クライアントの初期化を行うときに、ファイル `coldstart` がプロトタイプ・コールド・スタート・ファイルとして使用されるようにします。この `coldstart` ファイルは、既に NIS+ ネーム・スペースのクライアントであるマシンからコピーできます。最高の安全性を確保するため、管理者は、(`uuencode`(1C) を使って) ファイル `coldstart` を暗号化およびエンコードして、新しいマシンを起動する管理者にメールで送ります。新しい管理者は、受け取ったものを (`uudecode` を使って) デコードし、暗号化解除してから、`nisinit` コマンドにより、NIS+ クライアント・マシンを初期化するためにこのファイルを使用します。`coldstart` ファイルが同じドメイン内の別のクライアントからのものであれば、`nisinit` コマンドを安全な方法でスキップし、ファイルは、`/var/nis` ディレクトリーの `/var/nis/NIS_COLD_START` にコピーすることができます。

#### **-H** `hostname`

ホスト `hostname` を、トラステッド NIS+ サーバーとして連絡するように指定します。`nisinit` コマンドは、NETPATH 環境変数の各トランスポートごとに繰り返され、そのマシン

上の `rpcbind` との接続を試みます。このホスト名は、ネーム・サービスを実行しなくても、クライアントから読み取り可能でなければなりません。IP ネットワークの場合、これは、`nisinit` が起動されたときに、このホストの `/etc/hosts` 内に 1 つのエントリがなければならぬことを意味します。

- B `nisinit` コマンドが、IP ブロードキャストを使用して、ローカル・サブネット上の NIS+ サーバーを見つけるように指定します。NIS+ サーバーを実行しているどのマシンでも応答できます。応答したサーバーがその組織のネーム・スペースのサーバーであるという保証はされません。このフラグを使用する場合、システム管理者に、そのサーバーと、サービスの対象となっているドメインが有効であるか確認するようにお勧めします。バインド情報は、`nisshowcache` コマンドを使って標準出力に書き出すことができます。

**注:** `nisinit -c` は、このクライアントからの NIS+ ネーム・スペースのナビゲーションを使用可能にするだけです。NIS+ をユーザーのネーム・サービスにするには、そのことを反映するようにファイル `/etc/nsswitch.conf` を修正します。

-kkey\_domain

`root` のクレデンシャルが格納されているドメインを指定します。それを指定しない場合、システムのデフォルト・ドメインが想定されます。このドメイン名は、`/var/nis/NIS_COLD_START` ファイルを作成するために使用されません。

-pYIDINparent\_domain host...

ルート・サーバー上で `/var/nis/data/parent.object` を初期化して、このドメインが、そのサーバーの上にあるネーム・スペースの一部に含まれるようにします。ルート・サーバーだけが親オブジェクトを持つことができます。親オブジェクトは、NIS+ ルート上のネーム・スペースを記述します。これが分離されたドメインである場合は、このフラグを使用しないようにしてください。このフラグへの引数は、コマンドに対し、どのタイプのネーム・サーバーが NIS+ ドメイン上のドメインでサービスを行っているかを通知します。クライアントが NIS+ ネーム・スペースの外にある名前を解決しようとすると、このオブジェクトがエラー `NIS_FOREIGNNS` (ネーム・スペースの境界に達したことを示す) と一緒に戻されます。ネーム・レゾリューション・プロセスを継続するかどうかは、クライアントが決定します。

パラメーター "parent\_domain" は、そのタイプのドメインで固有である、1 つの構文内の親ドメインの名前です。ドメイン・パラメーターの後に続くホスト名のリストは、親ドメインにサービスを行うホストの名前です。1 つの親ドメイン用に複数のサーバーがある場合、最初に指定されたホストがそのドメインのマスター・サーバーとなります。

Y 親ディレクトリーが NIS バージョン 2 ドメインであることを指定します。

D 親ディレクトリーが DNS ドメインであることを指定します。

N parent\_domain\_host...

親ディレクトリーが別の NIS+ ドメインであることを指定します。このフラグは、既存の NIS+ サブツリーをグローバル・ネームスペース内に接続する場合に有用です。

- r NIS+ ルート・サーバーとなるマシンを初期化します。このフラグは、ファイル `/var/nis/data/root.object` を作成し、そのファイルを初期化してこのマシンに関する情報を入れるようにします。このフラグは、`sysinfo(2)` システム・コールを使用して、デフォルト・ドメインの名前を検索します。

## 例

1. クライアントを初期化するには、次を使用します。

```
nisinit -c -B
nisinit -c -H hostname
nisinit -c -C filename
```

2. ルート・マスター・サーバーを使用するには、次を使用します。

```
nisinit -r
```

クライアントを初期化する

3. クライアントは、次の 3 つの方法で初期化できます。

- ホスト名によって
- ブロードキャストによって
- コールド・スタート・ファイルによって

**注:** これらの方法にはそれぞれに異なる前提条件と関連タスクがあります。例えば、ホスト名によってクライアントを初期化する場合、そのクライアントの `/etc/hosts` ファイルが、使用するホスト名をリストし、`nsswitch.conf` ファイルが、`hosts` 行の最初の選択として `files` を指定してからでないとできません。前提条件と関連タスクなど、各メソッドについての詳細な説明は、NIS+ クライアントの初期化に記載されています。以下は、`nisinit` コマンドを使用する場合のステップの要約です。

4. ホスト名によってクライアントを初期化するには、`-c` オプションと `-H` オプションを使用し、クライアントがそのコールド・スタート・ファイルを入手する、入手元サーバーの名前を組み込みます。

```
nisinit -c -H hostname
```

5. コールド・スタート・ファイルによってクライアントを初期化するには、`-c` オプションと `-C` オプションを使用し、コールド・スタート・ファイルの名前を指定します。

```
nisinit -c -C filename
```

6. ブロードキャストによってクライアントを初期化するには、`-c` オプションと `-B` オプションを使用します。

```
nisinit -c -B
```

ルート・マスター・サーバーを初期化する

7. ルート・マスター・サーバーを初期化するには、`nisinit -r` コマンドを使用します。

```
nisinit -r
```

## ファイル

`/var/nis/NIS_COLD_START`

このファイルには、サーバー、そのトランスポート・アドレス、およびマシンのデフォルト・ドメインでサービスを行うセキュア RPC 公開鍵のリストが入っています。

`/var/nis/data/root.object`

このファイルは、NIS+ ネーム・スペースのルート・オブジェクトを記述します。これは、`nis_modify()` インターフェースを使って許可クライアントが変更できる、標準 XDR エンコード済み NIS+ ディレクトリーです。

`/var/nis/data/parent.object`

このファイルは、論理的に NIS+ ネーム・スペース上にあるネーム・スペースを記述します。最も一般的な親オブジェクトの型は DNS オブジェクトです。このオブジェクトは、そのドメインのサーバーに関する連絡情報を含んでいます。

`/etc/hosts`

インターネット・ホスト・テーブル。

## 関連情報

`nisclient` コマンド、および `nisshowcache` コマンド。

---

## nisln コマンド

### 目的

NIS+ オブジェクトとテーブル・エントリー間のシンボリック・リンクを作成します。

### 構文

```
nisln [ [ -L ] [ -D ] [source] [target] ]
```

### 説明

`nisln` コマンドは、オブジェクトとオブジェクトの間、またはオブジェクトとテーブル・エントリーの間をリンクします。NIS+ 管理コマンドはすべて、**-L** フラグを受け入れます。このフラグは、それらのコマンドに、NIS+ オブジェクト間のリンクの後に続くように指示します。

別のオブジェクトまたはエントリーとのリンクを作成するには、ソース・オブジェクト (すなわち、もう一方のオブジェクトまたはエントリーを指すオブジェクト) に対する修正権が必要です。

#### 注:

1. リンクは、テーブル・エントリーを起点とする場合には作成できません。
2. `cred` テーブルはリンクしてはなりません。各 `org_dir` ディレクトリーには、それぞれ独自の `cred` テーブルがあります。他の `org_dir cred` テーブルとのリンクは使用しないでください。

### フラグ

- |           |   |
|-----------|---|
| <b>-L</b> | リンクの後に続きます。 <b>source</b> 自体がリンクである場合、新しいリンクはそれにはリンクされませんが、そのリンクの起点となるソースにはリンクされます。 |
| <b>-D</b> | リンクされたオブジェクトに異なるデフォルト・セットを指定します。デフォルトについては、作成時にデフォルト以外のセキュリティー値を指定のセクションに説明があります。   |

### 例

オブジェクト間のリンクを作成するには、両方のオブジェクト名を指定します。この場合、最初に `source` を指定し、次に `target` を指定します。オブジェクトとエントリー間のリンクを作成するには、索引付きの名前を使用します。

```
nisln source-object target-object
nisln [column=value,...],tablename target-object
```

---

## nislog コマンド

### 目的

`nislog` コマンドは、トランザクション・ログの内容を表示します。

## 構文

**nislog** [ **-h** *num* | **-t** *num* ] [ **-v** ] [*directory*]...

## 説明

**nislog** コマンドは、トランザクション・ログの内容を表示します。

各トランザクションは、そのトランザクションに特定の部分と、オブジェクト定義のコピー部分の 2 つの部分から構成されます。

次の例は、**wiz.com**. ディレクトリーが最初に作成されたときに作られたトランザクション・ログ・エントリーを示しています。XID はトランザクション ID を示します。

```
rootmaster# /usr/sbin/nislog -h 1
NIS Log printing facility.
NIS Log dump:
  Log state : STABLE
  Number of updates      : 48
  Current XID            : 39
  Size of log in bytes   : 18432
  ***UPDATES***@@@@@@@@@@@@TRANSACTION@@@@@@@@@@@@#00000,
  XID : 1
  Time      : Wed Nov 25 10:50:59 1992
  Directory : wiz.com.
  Entry type : ADD Name
  Entry timestamp : Wed Nov 25 10:50:59 1992
  Principal   : rootmaster.wiz.com.
  Object name  : org_dir.wiz.com.
  .....Object.....
  Object Name : org_dir
  Owner       : rootmaster.wiz.com.
  Group       : admin.wiz.com.
  Domain      : wiz.com.
  Access Rights : r---rmcdr---r---
  Time to Live : 24:0:0
  Object Type  : DIRECTORY
  Name : `org_dir.wiz.com.'
  Type: NIS
  Master Server : rootmaster.wiz.com.
  .
  .
  .....
  @@@@@@@@@@@@@@@@@TRANSACTION@@@@@@@@@@@@@@@@
  #00000, XID : 2
```

## フラグ

- h** *num* ログのヘッド (先頭) からトランザクションを表示します。数字を省略すると、表示は最初のトランザクションから開始されます。数字 0 を入力すると、ログ・ヘッダーだけが表示されます。
- t** *num* ログの終わり (テール) から逆方向にトランザクションを表示します。数字を省略すると、表示は最後のトランザクションから開始されます。数字 0 を入力すると、ログ・ヘッダーだけが表示されます。
- v** 詳細モード

---

## nisls コマンド

### 目的

NIS+ ディレクトリーの内容をリストします。

## 構文

**nisls** [ **-d** ] [ **-g** ] [ **-l** ] [ **-L** ] [ **-m** ] [ **-M** ] [ **-R** ] [ *Directory...* ]

## 説明

**nisls** コマンドは、NIS+ ディレクトリーにあるパラメーターに指定された各ディレクトリーの内容を、標準出力に書き出します。 *Directory* がディレクトリーにない他のいずれかの NIS+ オブジェクトを指定している場合、**nisls** は、単にオブジェクトの名前をエコーするだけです。パラメーターとしてディレクトリーが指定されていない場合、検索パス内の最初のディレクトリー (デフォルト) がリストされます (**nisdefaults** を参照)。

## フラグ

- d** NIS+ ディレクトリーを、その内容をリストする代わりに、他の NIS+ オブジェクトと同様に扱います。
- g** **-l** フラグを使用して長形式でリストするときに、オーナーでなくグループ・オーナーを表示します。
- l** 長形式でリストします。**-l** フラグは、*Directory* に関する追加情報 (そのタイプ、作成時間、オーナー、アクセス権など) を表示します。
- L** リンクの後に続くように示します。*Directory* が実際に 1 つのリンクを指している場合、リンク・オブジェクトまで、そのリンクの後に続きます。
- m** **-l** フラグを使用して長形式で内容をリストするときに、作成時間の代わりに修正時間を表示します。
- M** 指名されたディレクトリーのマスター・サーバーが **nisls** コマンドの標準出力を戻すことを指定します。**-M** フラグを使用すると、最新の情報が確実にリストされます。
- R** ディレクトリーを再帰的にリストします。**-R** フラグは、*Directory* に指定されたディレクトリーに含まれる各サブディレクトリーの内容を表示します。

## 環境

**NIS\_PATH** NIS+ ディレクトリー名が完全修飾でない場合、オブジェクトが見つかるまで、指定された各ディレクトリーを検索します (**nisdefaults** を参照)。

## 終了状況

- 0 正常終了。
- 1 エラーが発生しました。

## 例

1. **org.com.** の内容 (そのサブディレクトリーなど) を短形式でリストするには、次のように入力します。  
`nisls -R org.com.`
2. **rootmaster.org.com.** に関する詳細情報 (最終変更時を含む) を表示するには、次のように入力します。  
`nisls -lm rootmaster.org.com.`

## 関連情報

**nisdefaults** コマンド、 **nisgrpadm** コマンド、 **nismatch** コマンド、および **nistbladm** コマンド。



---

## nismatch コマンド

### 目的

NIS+ テーブルを検索するためのユーティリティ。

### 構文

```
nismatch [ -A ] [ -c ] [ -h ] [ -M ] [ -o ] [ -P ] [ -v ]
```

### 説明

コマンド **nisgrep** は、単純なテキスト一致ではなく、検索基準における正規表現を受け入れる機能に関して、**nismatch** コマンドと異なっています。

**nisgrep** はコールバック機能を使用するので、テーブル作成時に特別に検索可能にされたカラムだけを検索するには制限されていません。これによって、**nismatch** に比べて柔軟性はありますが、速度は遅くなります。

**nismatch** ではサーバーが検索を行います。それに対し、**nisgrep** ではサーバーがすべての読み取り可能エントリーを戻してから、クライアントがパターン・マッチングを行います。

どちらのコマンドも、パラメーター **tablename** は、検索されるテーブルの NIS+ 名です。カラム名を指定せずに 1 つのキーまたはキー・パターンだけを指定すると、そのキーまたはキー・パターンは最初のカラムの検索時に適用されます。特定の名前が付いたカラムは、その構文を使用して検索できます。

複数のカラムを検索するときは、すべてのカラムで一致するエントリーだけが戻されます。これは、論理結合演算と同じことです。**nismatch** は検索基準の追加形式を受け入れます。これはその形式の NIS+ 索引付きの名前です。

### フラグ

- A そのテーブル内のデータと、初期テーブルの連結パス内にあるテーブルの全データを戻します。
- c 検索基準と一致するエントリー数のカウントだけを出力します。
- h テーブルのカラムの名前が入っている突き合わせエントリーの前に、ヘッダー行を表示します。
- M マスター・サーバーのみ。指名されたデータのマスター・サーバーにルックアップを送信します。これによって、最新の情報を表示することが保証されますが、場合によってはマスター・サーバーが使用中であるという欠点があります。
- o 突き合わせ NIS+ オブジェクトの内部表現を表示します。
- P 連結パスの後に続きます。初期検索が失敗であった場合、ルックアップをテーブルの連結パスの後に続けるように指定します。
- v 突き合わせエントリーを表示するときに、バイナリー・データの出力を抑止しません。このオプションを使用しない場合、バイナリー・データは文字列 `*s-1BINARYs0*` として表示されます。
  1. 0 - いくつかのエントリーが正常に突き合わせられました。
  2. 1 - テーブルの検索が正常に行われ、一致は見つかりませんでした。
  3. 2 - エラー状態が起りました。エラー・メッセージも出力されます。

### 例

1. この例では、**zotz.com.domain** の **org\_dir** サブディレクトリー内のテーブル **passwd** を検索します。ユーザー名が **skippy** というエントリーを戻します。

この例では、すべての作業がサーバー上で行われます。

```
nismatch\ name=skippy\ passwd.org_dir.zotz.com.
```

2. この例は上記の例と類似していますが、**nisgrep** を使用して、テーブル **passwd** 内で **ksh** (1) または **csh** (1) を使用しているすべてのユーザーを見つける、という点が異なります。

```
nisgrep\ 'shell=[ck]sh'\ passwd.org_dir.zotz.com.
```

3. **NIS\_PATH** - この変数が設定されていて、**NIS+** テーブル名が完全修飾でない場合、指定された各ディレクトリーは、そのテーブルが見つかるまで検索されます (**nisdefaults**、**niscat**、**nisls**、および **nistbladm** を参照)。

## 関連情報

**nisgrep** コマンド、**nisdefaults** コマンド、**niscat** コマンド、**nisls** コマンド、および **nistbladm** コマンド。

---

## nismkdir コマンド

### 目的

ルート・ディレクトリーでない **NIS+** ディレクトリーを作成します。

### 構文

```
nismkdir [ -D Defaults ] [ -m MasterHost | -s ReplicaHost ] DirName
```

### 説明

**nismkdir** コマンドは、既存ドメイン内にサブディレクトリーを作成します。また、複製ディレクトリーも作成します。フラグを使用しない場合、**nismkdir** コマンドは、その親ディレクトリーと同じマスター・サーバーとレプリカ・サーバーを使ってサブディレクトリーを作成します。さらに、**nismkdir** コマンドは、既存のディレクトリーにレプリカを追加できます。

**NIS+** ディレクトリーにサービスを行うホストは、サービスの対象となるディレクトリー上のディレクトリーにある **NIS+** クライアントでなければなりません。唯一の例外は、同じ **NIS+** ディレクトリーに対してクライアントとサーバーの両方の働きをするルート **NIS+** サーバーです。

ホストのデフォルト・ドメインが、**nismkdir** コマンドが実行されるドメインでない場合、**-s** または **-m** フラグを指定したパラメーター内に指定されたホスト名は完全修飾でなければなりません。

**注:** **org\_dir** および **group\_dir** によって指名されたディレクトリーで構成される **NIS+** ドメインを作成する場合は、**nisserver** コマンドを使用する必要があります。

### フラグ

**-m MasterHost**

*DirName* パラメーターによって指名されたディレクトリーがまだ存在しない場合、**-m** フラグは、そのマスター・サーバーとして *MasterHost* を指定した新しいディレクトリーを作成します。 *DirName* によって指名されたディレクトリーがある場合、*MasterHost* パラメーターによって指名されたホストがそのマスター・サーバーとなります。

**注:** ディレクトリーを作成する場合は、そのドメインのマスター・サーバー上の親ディレクトリーに対する作成権が必要です。

## **-s** *ReplicaHost*

ルートでない NIS+ ディレクトリーとそのマスター・サーバーを、既存のシステムに追加します。また、**-s** フラグは、新しいレプリカ・サーバーを既存のディレクトリーに割り当てることができます。 *DirName* が既に存在する場合、**nismkdir** コマンドはそれを再作成せず、その代わりに、単に新しいレプリカ・サーバーをその既存のディレクトリーに割り当てます。

**-s** フラグを起動した後は、レプリカ・サーバーが追加または割り当てられたディレクトリー上のマスター・サーバーから、**nisping** コマンドを実行する必要があります。各ディレクトリーに **nisping** コマンドをそのマスター・サーバーの **cron** ファイルに組み込んで、そのディレクトリーが、更新されるまでに、少なくとも 24 時間に 1 回は PING されるようにしてください。

### 注:

1. サーバーは、ルート・ドメインに所属していない限り、その親ドメインをサポートするために割り当ててはできません。
2. **nismkdir** コマンドは常にマスター・サーバーで実行します。レプリカ・サーバーでは決して **nismkdir** を実行しないでください。レプリカ・サーバーで **nismkdir** を実行すると、マスターとレプリカの間で通信上の問題が起こります。

## **-D** *Defaults*

新しいディレクトリーに異なるデフォルト・セットを指定します。 *defaults* 文字列は、コロンで区切られた一連のトークンです。これらのトークンは、汎用オブジェクト属性に使用するデフォルト値を表します。

### **ttl=***Time*

**nismkdir** コマンドによって作成されたオブジェクトのデフォルトの存続時間を設定します。値 *Time* は、**nischttl** コマンドによって定義されたフォーマットで指定します。デフォルト値は 12h (12 時間) です。

### **owner=***Ownername*

NIS+ プリンシパル *Ownername* が、作成されたオブジェクトを所有するように指定します。この値のデフォルトは、そのコマンドを実行しているプリンシパルです。

### **group=***Groupname*

グループ *Groupname* が、作成されたオブジェクトのグループ・オーナーとなるように指定します。デフォルト値は NULL です。

### **access=***Rights*

作成されたオブジェクトに関して認可される 1 組のアクセス権限を指定します。値 *Rights* は、**nischmod** コマンドによって定義されたフォーマットで指定する必要があります。デフォルト値は `—rmcdr—r—` です。

## 環境

### **NIS\_DEFAULTS**

NIS+ 標準デフォルトをオーバーライドするデフォルト文字列が入っています。 **-D** フラグを起動する場合、これらの値は、**NIS\_DEFAULTS** 変数と標準デフォルトの両方をオーバーライドします。

### **NIS\_PATH**

NIS+ ディレクトリー名が完全修飾でない場合、そのディレクトリーが見つかるまで、指定されたすべてのディレクトリーを検索します (**nisdefaults** を参照)。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0 正常終了。
- 1 エラーが発生しました。

## 例

1. abc.com. ディレクトリーと同じマスターとレプリカを共用する abc.com. ドメインのもとで、新しいディレクトリー bar を作成するには、次のように入力します。

```
nismkdir def.abc.com.
```

2. abc.com. ドメインのもとで複製されていない新しいディレクトリー def.abc.com. を作成するには、次のように入力します。

```
nismkdir \-m myhost.abc.com.\ def.abc.com.
```

3. def.abc.com. ディレクトリーのレプリカ・サーバーを追加するには、次のように入力します。

```
nismkdir \-s replica.abc.com.\ def.abc.com.
```

## ファイル

## 関連情報

**nischmod** コマンド、 **nisdefaults** コマンド、 **nisls** コマンド、 **nisrmdir** コマンド、 および **nissserver** コマンド。

---

## nismkuser コマンド

### 目的

新しい NIS+ ユーザー・アカウントを作成します。

### 構文

```
nismkuser [ Attribute=Value ... ] Name
```

### 説明

**nismkuser** コマンドは、NIS+ ドメイン内に NIS+ ユーザー・エントリーを作成します。 *Name* パラメーターは、固有の 8 バイト以下の文字列でなければなりません。ユーザー名には **ALL** キーワードまたは **default** キーワードは使用できません。デフォルトでは、**nismkuser** コマンドは標準のユーザー・アカウントを作成します。管理ユーザー・アカウントを作成するには、**-a** フラグを指定します。

**注:** **nismkuser** コマンドを使用して NIS+ グループにユーザーを追加することはできません。この機能を実行するには、**nisgrpadm** コマンドを使用してください。

**nismkuser** コマンドによって、ユーザー作成時に NIS+ ユーザー・パスワードの入力が可能になります。ユーザー作成時にパスワードを入力しない場合、NIS+ ユーザーの LOCAL および DES cred がパスワード nisplus を指定して作成されます。後で、パスワードは、**passwd** コマンドによってセットまたはリセットできます。新しいアカウントは使用不可にならず、**nismkuser** コマンドの完了後もアクティブです。

**注:**

1. このコマンドにより、ユーザーは NIS+ ユーザーの「ホーム」ディレクトリーを設定できますが、そのディレクトリーがまだ存在しない場合、実際の物理ディレクトリーは作成されません。
2. ユーザーを追加するには、その前に、新規ユーザーと同じ gid (グループ ID) が指定されたグループを `group.org_dir` に入れておく必要があります。 **nismkuser** のデフォルト gid は 1 です。

このコマンドを (NIS+ 管理域のもとで) 実行するために、Web-Based System Manager ユーザー・アプリケーションまたは System Management Interface Tool (SMIT) が使用できます。

## ユーザー名作成に関する制約事項

ログインの不整合を防ぐには、ユーザー名を全部英大文字にはしません。 **nismkuser** コマンドはマルチバイトのユーザー名をサポートしますが、ユーザー名は、POSIX ポータブル・ファイル名文字セットを含む文字に制限するようにお勧めします。

ユーザー・データベースが絶対に破壊されないようにするには、ユーザーの命名をするときに注意する必要があります。ユーザー名は、- (ダッシュ)、+ (正符号)、@ (アットマーク)、または ~ (ティルド) で始めてはなりません。ユーザー名には **ALL** キーワードまたは **default** キーワードは使用できません。さらに、ユーザー名文字列の中では、次のどの文字も使用してはなりません。

. ドット  
: コロン  
" 二重引用符  
# ポンド記号  
, コンマ  
= 等号  
\ バックスラッシュ  
/ スラッシュ  
? 疑問符  
' 単一引用符  
' 逆引用符

**重要:** 既存の NIS+ クライアントまたはサーバー名と同じ名前の NIS+ ユーザーを作成することは認められません。

最後に、*Name* パラメーターには、スペース、タブ、または改行文字を含めることはできません。

## パラメーター

*Attribute=Value* ユーザー属性を初期化します。有効な属性および値については、 **chuser** コマンドを参照してください。  
*Name* 固有の 8 バイト以下の文字列を指定します。

## 有効なパラメーター

**nismkuser** を使用すると、管理者が、**mkuser** コマンドの場合と同じ属性およびパラメーターを入力できます。ただし、**nismkuser** コマンドでは次のパラメーターのみが使用されます (これ以外は無視され、エラーとは見なされません)。

**id, pgrp, gecos, shell, home, minage, maxage, maxexpired, password, pldwarntime**

## セキュリティー

アクセス制御: このコマンドは、実行 (x) アクセス権を root ユーザーとセキュリティー・グループのメンバーだけに与えます。このコマンドは、トラステッド・コンピューティング・ベース (TCB) 内のプログラムとしてインストールします。このコマンドは、**setuid** (SUID) ビットが設定されている root ユーザーが所有しなければなりません。

監査イベントは次のとおりです。

イベント	情報
USER_Create	ユーザー

## 例

1. **/usr/lib/security/nismkuser.default** ファイル内のデフォルト値を使って davis ユーザー・アカウントを作成するには、次のように入力します。

```
nismkuser davis
```

2. davis ユーザー・アカウントを作成して、**su** 属性を **false** の値に設定するには、次のように入力します。

```
nismkuser su=false davis
```

## ファイル

**/usr/bin/nismkuser** **nismkuser** コマンドが入っています。

## 関連情報

**chfn** コマンド、**chgroup** コマンド、**chgrpmem** コマンド、**chsh** コマンド、**chuser** コマンド、**lsgroup** コマンド、**lsuser** コマンド、**mkggroup** コマンド、**passwd** コマンド、**pwdadm** コマンド、**rmgroup** コマンド、**rmuser** コマンド、**setgroups** コマンド、**setsenv** コマンド。

ユーザーの識別と認証、任意アクセス制御、トラステッド・コンピューティング・ベース、および監査の詳細については、セキュリティーのネットワークの保護のセクションを参照してください。

管理ロールの詳細については、セキュリティーのユーザー、ロール、およびパスワードのセクションを参照してください。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

---

## nisping コマンド

### 目的

レプリカ・サーバーを PING し、それらのサーバーに、すぐに更新をマスター・サーバーに依頼するように指示します。レプリカが応答すると、**nisping** は、root マスター・サーバーの **niscachemgr** キャッシュ・ファイル **/var/nis/NIS\_SHARED\_DIRCACHE** に入っている、そのレプリカのエントリーを更新します。

注: これらのレプリカは、通常数分待ってからこの要求を実行します。

## 構文

### 最後の更新の時間を表示する

**nisping** [ **-u** *domain* ]

### レプリカを PING する

**nisping** [ **-H** *hostname* ] [*domain*]

### ディレクトリーのチェックポイントを指定する

**nisping** [ **-C** *hostname* ] [*domain* ]

## 説明

PING を行う前に、コマンドは、各レプリカから受信した最後の更新の時間を調べます。その時間がマスターによって送信された最後の更新時間と同じ場合、このコマンドはレプリカを PING しません。

**nisping** コマンドは、ディレクトリーのチェックポイントを指定することもできます。この指定には、ディレクトリー内の各サーバー (マスターも含む) へ、ドメインのトランザクション・ログからディスク上に、そのサーバーの情報を更新することを通知することが含まれます。

## フラグ

**-u** *domain*

最後の更新の時間を表示します。どのサーバーにも PING は送られません。

**-H** *hostname*

ホスト **hostname** のみに PING が送信され、更新時間についてのチェックが行われるか、チェックポイントが指定されます。

**-C** *hostname*

各サーバーへ PING を送信するのではなく、チェックポイントを指定するための要求を送信します。これらのサーバーは、すべてのトランザクションを継続的ストレージにコミットするようにスケジュールします。

## 例

### 最終更新の時間を表示する

**-u** フラグを使用します。このフラグは、別のドメイン名を指定しない限り、ローカル・ドメインのマスターとレプリカの更新時間を表示します。PING は行いません。

```
/usr/lib/nis/nisping -u [domain]
```

次に例を示します。

```
rootmaster# /usr/lib/nisping -u org_dir
Last updates for directory wiz.com.:
Master server is rootmaster.wiz.com.
    Last update occurred at Wed Nov 25 10:53:37 1992
Replica server is rootreplica1.wiz.com.
    Last update seen was Wed Nov 25 10:53:37 1992
```

### レプリカを PING する

ドメイン内のレプリカ全部、または特定の 1 つだけを PING できます。すべてのレプリカを PING するには、オプションを指定せずに次のコマンドを使用します。

```
/usr/lib/nis/nisping
```

ローカル・ドメイン以外のドメイン内のレプリカ全部を PING するには、ドメイン名を付加します。



```
/usr/lib/nis/nisping domainname
```

次に、ローカル・ドメイン **wiz.com** のすべてのレプリカを PING する場合の例を示します。

```
rootmaster# /usr/lib/nis/nisping org_dir
Pinging replicas serving directory wiz.com.:
Master server is rootmaster.wiz.com.
    Last update occurred at Wed Nov 25 10:53:37 1992
Replica server is rootreplica1.wiz.com.
    Last update seen was Wed Nov 18 11:24:32 1992

Pinging ... rootreplica1.wiz.com.
```

更新時間が異なっていたため、そのまま PING を行います。この時間が同じであったら、PING を送信していません。

1 つの指定ホスト上の全ディレクトリー内のテーブルをすべて PING することもできます。特定のホストの全ディレクトリー内のテーブルをすべて PING するには、**-a** フラグを使用します。

```
/usr/lib/nis/nisping -a hostname
```

## ディレクトリーのチェックポイントを指定する

ディレクトリーのチェックポイントを指定するには、**-C** フラグを使用します。

```
/usr/lib/nis/nisping -C directory-name
```

1 つのドメインをサポートするすべてのサーバー (マスターも含む) は、自らの情報をその **.log** ファイルからディスクに転送します。これにより、ログ・ファイルが消去され、ディスク・スペースが解放されます。サーバーは、チェックポイントを指定しているとき、サービスの要求には応じますが更新には使用できません。

以下は、**nisping** 出力の例です。

```
rootmaster# /usr/lib/nis/nisping -C
Checkpointing replicas serving directory wiz.com. :
Master server is rootmaster.wiz.com.
    Last update occurred at Wed May 25 10:53:37 1995
Master server is rootmaster.wiz.com.
checkpoint has been scheduled with rootmaster.wiz.com.
Replica server is rootreplica1.wiz.com.
    Last update seen was Wed May 25 10:53:37 1995
Replica server is rootreplica1.wiz.com.
checkpoint has been scheduled with rootmaster.wiz.com.
```

---

## nispopulate コマンド

### 目的

NIS+ ドメイン内の NIS+ テーブルに転送します。

### 構文

```
nispopulate -Y [ -x ] [ -f ] [ -n ] [ -u ] [ -v ] [ -S 0 | 2 ] [ -l network_passwd ] [ -d NIS+_domain ]
-h NIS_server_host [ -a NIS_server_addr ] -y NIS_domain [ table ] ...
```

```
nispopulate -F [ -x ] [ -f ] [ -u ] [ -v ] [ -S 0 | 2 ] [ -d NIS+_domain ] [ -l network_passwd ] [ -p
directory_path ] [ table ] ...
```

```
nispopulate -C [ -x ] [ -f ] [ -v ] [ -d NIS+_domain ] [ -l network_passwd ] [ hosts | passwd ]
```

## 説明

**nispopulate** コマンドは、指定されたドメイン内の NIS+ テーブルに、それに対応するファイルまたは NIS マップから転送する場合に使用できます。 **nispopulate** コマンドは、そのテーブルが、 **nisserver** コマンドまたは **nissetup** コマンドを使って作成されたものと想定します。

*table* 引数は、標準の名前と非標準 *key-value* タイプのテーブルを受け入れます。 *key-value* タイプのテーブルについて詳しくは、 **nisaddent** を参照してください。 *table* 引数が指定されない場合、 **nispopulate** は自動的に各標準テーブルに転送します。これらの標準 (デフォルト) テーブルとは、 **auto\_master**、 **auto\_home**、 **ethers**、 **group**、 **hosts**、 **networks**、 **passwd**、 **protocols**、 **services**、 **rpc**、 **netmasks**、 **bootparams**、 **netgroup**、 **aliases**、 **shadow** です。

注: **shadow** テーブルは、ファイルから転送するときだけに使用します。 **nispopulate** が受け入れる非標準テーブルは、 *key-value* タイプのテーブルです。これらのテーブルは、最初に **nistbladm** コマンドを使って手動で作成しておく必要があります。

NIS マップから NIS+ テーブルに転送する場合は、最初の構文 (-Y) を使用します。 **nispopulate** コマンドは、 **ypxfr** コマンドを使用して、NIS マップを、NIS サーバーからローカル・マシン上の **/var/yp/NIS\_domain** ディレクトリーに転送します。その上で、それらのファイルを入力ソースとして使用します。

注: *NIS\_domain* は大文字小文字を区別します。そのディレクトリー用に十分なディスク・スペースを確保するようにしてください。

2 番目の構文 (-F) は、ローカル・ファイルから NIS+ テーブルに転送する場合に使用します。

**nispopulate** コマンドは、テーブル名と一致するファイルを、現在の作業ディレクトリー内または指定ディレクトリー内の入力ソースとして使用します。

**nispopulate** コマンドは、 **hosts** テーブルおよび **passwd** テーブルに転送するとき、自動的に、 **hosts** テーブルと **passwd** テーブル内でそれぞれ定義された全ユーザーおよびホストの NIS+ クレデンシャルを作成します。これらのクレデンシャルを作成するには、ネットワーク・パスワードが必要です。このネットワーク・パスワードは、新しいユーザーとホストの秘密鍵を暗号化するために使用されます。このパスワードは、 **-i** フラグを使用して指定できますが、デフォルト・パスワードの **nisplus** を使用する場合があります。この **nispopulate** は、クレデンシャル・テーブル内の既存のクレデンシャル・エントリーを上書きしません。クレデンシャル・テーブルのエントリーを上書きするには、 **nisclient** を使用します。これは、ユーザーには LOCAL と DES の両方のクレデンシャルを作成し、ホストには DES クレデンシャルのみを作成します。自動クレデンシャル作成を使用不可にする場合は、 **-S 0** フラグを指定します。

3 番目の構文 (-C) は、レベル 2 認証 (DES) の NIS+ クレデンシャル・テーブルに、指定されたドメインのパスワードとホストのテーブルから転送する場合に使用します。この操作での有効な *table* 引数は、 **passwd** と **hosts** です。この引数が指定されないと、入力ソースとして **passwd** と **hosts** の両方が使用されます。

**nispopulate** が以前に **-S 0** フラグを指定して使用されている場合、そのホストまたはユーザーにはクレデンシャルは追加されていません。それ以降、そのサイトで全ユーザーおよびホストのクレデンシャルを追加することになった場合、この (-C) フラグを使用してクレデンシャルを追加できます。

**nispopulate** コマンドは、通常、ディレクトリー **/tmp** 内に一時ファイルを作成します。環境変数 **TMPDIR** を任意で選択したディレクトリーに設定すれば、別のディレクトリーを指定できます。 **TMPDIR** が有効なディレクトリーでない場合、 **nispopulate** は **/tmp** を使用します。

## フラグ

- a** *NIS\_server\_addr*  
NIS サーバーの IP アドレスを指定します。このフラグは、**-Y** フラグと一緒にしか使用されません。
- C**  
DES 認証 (セキュリティー・レベル 2) を使用して、パスワードとホストのテーブルから NIS+ クレデンシャル・テーブルに転送します。
- d** *NIS+\_domain.*  
NIS+ ドメインを指定します。デフォルトはローカル・ドメインです。
- F**  
ファイルから NIS+ テーブルに転送します。
- f**  
スクリプトに、確認のプロンプトを出さずに NIS+ テーブルに転送するよう強制します。
- h** *NIS\_server\_host*  
NIS マップのコピー元である NIS サーバー・ホスト名を指定します。これは、**-Y** フラグと一緒にしか使用されません。このホストは、NIS+ **hosts** テーブルまたは **/etc/hosts** ファイルに既に存在しているものでなければなりません。ホスト名が定義されない場合、スクリプトが、その IP アドレスを入力するプロンプトを出します。または **-a** フラグを使って、そのアドレスを手動で指定することもできます。
- l** *network\_passwd*  
NIS+ クレデンシャル・テーブルを転送するためのネットワーク・パスワードを指定します。これは、**hosts** テーブルおよび **passwd** テーブルを転送しているときのみ使用します。デフォルト・パスワードは **nisplus** です。
- n**  
**/var/yp/NISdomain** ディレクトリー内にローカル NIS マップが既にある場合、それらの上書きをしません。デフォルトは、ローカル **/var/yp/NISdomain** ディレクトリー内の既存の NIS マップを上書きすることです。これは、**-Y** フラグと一緒にしか使用されません。
- p** *directory\_path*  
ファイルが格納されるディレクトリーを指定します。これは、**-F** フラグと一緒にしか使用されません。デフォルトは、現在の作業ディレクトリーです。
- S 0 | 2**  
NIS+ クライアントの認証レベルを指定します。レベル 0 は認証されていないクライアントのレベルで、指定されたドメイン内のユーザーおよびホストのクレデンシャルは作成されません。レベル 2 は認証された (DES) クライアントのレベルで、指定されたドメイン内のユーザーおよびホストの DES クレデンシャルが作成されます。デフォルトは、レベル 2 認証 (DES) を指定してセットアップすることです。レベル 0 認証の場合は、**-C** フラグを指定して **nispopulate** コマンドを実行する必要はありません。
- u**  
ファイルまたは NIS マップから NIS+ テーブルを更新 (すなわち、追加、削除、修正) します。このフラグは、変更の数が少ししかない場合に、NIS+ テーブルを最新のものにするために使用してください。デフォルトは、既存のエントリーを削除せずに NIS+ テーブルへ追加することです。また、**e /var/yp** ディレクトリー内の既存マップから NIS+ テーブルを更新する場合は、**-n** フラグを参照してください。
- v**  
詳細モードでスクリプトを実行します。

- x** 「エコー」モードをオンにします。スクリプトは、実行しようとしていたコマンドだけを出力します。実際には、それらのコマンドは実行されません。デフォルトはオフです。
- Y** NIS+ テーブルに NIS マップから転送します。
- y NIS\_domain** NIS マップにコピー元である NIS ドメインを指定します。これは、**-Y** フラグと一緒にしか使用されません。デフォルト・ドメイン名はローカル・ドメイン名と同じです。

## 例

1. ドメイン xyz.ibm.com. 内の全 NIS+ 標準テーブルに、入力ソースとして yp.ibm.com ドメイン (ただし、ホスト yp\_host は yp.ibm.com の YP サーバー) の NIS マップから転送するには、次のように入力します。

```
/usr/lib/nis/nispopulate -Y -y yp.ibm.COM -h yp_host -d xyz.ibm.com.
```

2. NIS+ 標準テーブルのすべてを、上記に示す同じ NIS ドメインおよびホストから更新するには、次のように入力します。

```
/usr/lib/nis/nispopulate -Y -u -y yp.ibm.COM -h yp_host -d xyz.ibm.com.
```

3. ドメイン xyz.ibm.com. の **hosts** テーブルに、**/var/nis/files** ディレクトリー内のホスト・ファイルから、キー暗号化のネットワーク・パスワードとして somepasswd を使用して転送するには、次のように入力します。

```
/usr/lib/nis/nispopulate -F -p /var/nis/files -l somepasswd hosts
```

4. 自動的に NIS+ クレデンシャルを作成せずに、ドメイン xyz.ibm.com. 内の passwd テーブルに、**/var/nis/files** ディレクトリー内の passwd ファイルから転送するには、次のように入力します。

```
/usr/lib/nis/nispopulate -F -p /var/nis/files -d xys.ibm.com. -S 0 passwd
```

5. passwd テーブルに定義された全ユーザーについて、ドメイン xyz.ibm.com. 内のクレデンシャル・テーブルに転送するには、次のように入力します。

```
/usr/lib/nis/nispopulate -C -d xys.ibm.com. passwd
```

6. 非標準 key-value タイプ NIS+ テーブル private を、ファイル **/var/nis/files/private:** (nispopulate は、private.org\_dirkey-value タイプ・テーブルは既に作成されていると想定します) から作成し、それに転送するには、次のように入力します。

```
/usr/bin/nistbladm -D access=og=rmcd,nw=r \  
-c private key=S,nogw= value=,nogw= private.org.dir  
/usr/lib/nis/nispopulate -F -p /var/nis/files private
```

## ファイル

<b>/etc/hosts</b>	ローカル・ホスト名データベース
<b>/var/yp</b>	NIS (YP) ドメイン・ディレクトリー
<b>/var/nis</b>	NIS+ ドメイン・ディレクトリー

## 関連情報

**nistbladm** コマンド、**nisaddcred** コマンド、**nisaddent** コマンド、**nisclient** コマンド、**nisserver** コマンド、**nissetup** コマンド、**rpc.nisd** コマンド、**ypxfr** コマンド。

---

## nisrm コマンド

### 目的

ネーム・スペースから NIS+ オブジェクトを除去します。

### 構文

```
nisrm [ -i ] [ -f ] Obj_name...
```

### 説明

**nisrm** コマンドは、NIS+ ネーム・スペースから NIS+ オブジェクトを除去します。**nisrm** コマンドは、NIS+ マスター・サーバーが実行していなければ失敗します。

注: **nisrm** は、ディレクトリー (**nisrmdir** コマンドを参照) も、空でないテーブル (**nistbladm** を参照) も除去しません。

- i **nisrm** コマンドを対話モードで設定します。-i フラグを使用すると、**nisrm** コマンドは、指定されたオブジェクトを除去する前に確認を求めてきます。そのオブジェクト名が完全修飾でない場合は、-i フラグが強制的に使用され、他のオブジェクトが間違っ除去されないようにします。
- f **nisrm** コマンドを強制モードで設定します。**nisrm** が、必要な許可がないために失敗した場合、**nischmod** が起動され、除去が再度試みられます。**nisrm** は、失敗した場合でもエラー・メッセージは戻しません。

### 例

1. ネーム・スペースからオブジェクト xyz、abc、および def を除去するには、次のように入力します。

```
nisrm xyz abc def
```

### 環境

**NIS\_PATH** この変数が設定されていると、NIS+ オブジェクト名が完全修飾でない場合には、**nisrm** は、そのオブジェクトが見つかるまで示された各ディレクトリーを検索します。

### 終了状況

- 0 正常終了。
- 1 エラーが発生しました。

### 関連情報

**nischmod** コマンド、**nisdefaults** コマンド、**nisrmdir** コマンド、**nistbladm** コマンド、および **rm** コマンド。

---

## nisrmdir コマンド

### 目的

ネーム・スペースから NIS+ オブジェクトを除去します。

## 構文

```
nisrmdir [ -i ] [ -f ] [ -s Hostname ] Dirname
```

## 説明

**nisrmdir** コマンドは、既存の NIS+ ディレクトリーとサブディレクトリーを除去します。 **nisrmdir** コマンドは、レプリカがディレクトリーにサービスを行わないように除去することもできます。

**nisrmdir** コマンドは、ディレクトリー (パラメーター *Dirname* に示されている) を記述するオブジェクトを修正してから、各レプリカにそれを除去するように通知します。この通知が失敗であると、ディレクトリー・オブジェクトは、**-f** フラグが使用されていない限り、その元の状態に戻されます。

**nisrmdir** は、NIS+ マスター・サーバーが実行していなければ失敗します。

- i** **nisrmdir** コマンドを対話モードで設定します。 **-i** フラグを使用すると、 **nisrm** コマンドは、指定されたオブジェクトを除去する前に確認を求めてきます。 *Dirname* のディレクトリー名が完全修飾でない場合、 **-i** フラグが強制的に使用され、他のディレクトリーが間違っ除去されないようにします。
- f** **nisrm** コマンドを強制モードで設定します。 **-f** フラグは、 **nisrmdir** に、そのコマンドが影響を受けるレプリカ・サーバーに連絡できない可能性がある場合でも、継続させるようにします。このフラグは、レプリカがダウンしていて、除去の通知に回答できないことがわかっている場合に使用します。レプリカは、最終的にリブートされると、更新されたディレクトリー・オブジェクトを読み取り、 *Dirname* に関してはレプリカではなくなったことを認識し、そのディレクトリーのルックアップへの応答を停止します。
- s Hostname** サーバー *Hostname* を、ディレクトリー *Dirname* のレプリカとして除去するように指定します。 **-s** フラグが使用されていない場合、 *Dirname* のレプリカ・サーバーとマスター・サーバーがすべて除去され、そのディレクトリーもネーム・スペースから除去されます。
- 注:** 除去されたディレクトリーを保持していたファイルは、 */var/nis* ディレクトリー内の該当のファイルを手動で除去することによって整理することができます。

## 例

1. abc.com. ドメインのもとでディレクトリー xyz を除去するには、次のように入力します。  

```
nisrmdir xyz.abc.com.
```
2. ディレクトリー xyz.abc.com. にサービスを行うレプリカを除去するには、次のように入力します。  

```
nisrmdir -s replica.abc.com xyz.abc.com.
```
3. ネーム・スペースからのディレクトリー xyz.abc.com. の除去を強制するには、次のように入力します。  

```
nisrmdir -f xyz.abc.com.
```

## 環境

**NIS\_PATH** この変数が設定されていると、NIS+ ディレクトリー名が完全修飾でない場合には、 **nisrmdir** は、そのディレクトリーが見つかるまで示された各ディレクトリーを検索します。

## 終了状況

- 0 正常終了。  
1 エラーが発生しました。



## 関連情報

`nisdefaults` コマンドおよび `nisrm` コマンド。

---

## nisrmuser コマンド

### 目的

NIS+ ユーザー・アカウントを除去します。

### 構文

`nisrmuser` *Name*

### 説明

`nisrmuser` コマンドは、*Name* パラメーターによって識別された NIS+ ユーザー・アカウントを除去します。ユーザーについて定義されたすべての属性が除去されますが、ユーザーのホーム・ディレクトリーとユーザーが所有するファイルは除去されません。ユーザー名は、8 文字以内の文字列として既に存在していなければなりません。

管理ユーザーを除去できるのは root ユーザーだけです。管理ユーザーは、`/etc/security/user` ファイルに `admin=true` と設定されているユーザーのことです。

Web-based System Manager ユーザー・アプリケーションまたは System Management Interface Tool (SMIT) を使用すると、このコマンドを NIS+ 管理セクション内で実行できます。

### セキュリティ

アクセス制御: このコマンドは、実行 (x) アクセス権を root ユーザーとセキュリティ・グループのメンバーだけに与えます。このコマンドは、トラステッド・コンピューティング・ベース (TCB) 内のプログラムとしてインストールします。このコマンドは、`setuid` (SUID) ビットが設定されている root ユーザーが所有しなければなりません。

監査イベント:

イベント	情報
<code>USER_Remove</code>	ユーザー

### 例

1. ユーザー `davis` のアカウントとその属性をローカル・システムから除去するには、次のように入力します。

```
nisrmuser davis
```

### ファイル

`/usr/sbin/nisrmuser` `nisrmuser` コマンドが入っています。



## 関連情報

**chfn** コマンド、**chgrp** コマンド、**chsh** コマンド、**chgroup** コマンド、**chuser** コマンド、**ls** コマンド、**lsuser** コマンド、**mkgroup** コマンド、**mkuser** コマンド、**passwd** コマンド、**pwdadm** コマンド、**rmgroup** コマンド、**setgroups** コマンド、**setsenv** コマンド。

ユーザーの識別と認証、任意アクセス制御、トラステッド・コンピューティング・ベース、および監査の詳細については、セキュリティのネットワークの保護のセクションを参照してください。

管理ロールの詳細については、セキュリティのユーザー、ロール、およびパスワードのセクションを参照してください。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

---

## nisserver コマンド

### 目的

NIS+ サーバーをセットアップします。

### 構文

#### ルート・マスター・サーバーをセットアップする

```
/usr/lib/nis/nisserver -r [ -d Domain ] [ -f ] [ -g GroupName ] [ -l Password ] [ -v ] [ -x ] [ -Y ]
```

#### 非ルート・マスター・サーバーをセットアップする

```
/usr/lib/nis/nisserver -M -d Domain [ -f ] [ -g GroupName ] [ -h HostName ] [ -v ] [ -x ] [ -Y ]
```

#### レプリカ・サーバーをセットアップする

```
/usr/lib/nis/nisserver -R [ -d Domain ] [ -f ] [ -h HostName ] [ -v ] [ -x ] [ -Y ]
```

### 説明

**nisserver** コマンドは、レベル 2 セキュリティ (DES) で、ルート・マスター、非ルート・マスター、およびレプリカの各 NIS+ サーバーをセットアップするために使用するシェル・スクリプトです。

新しいドメインをセットアップするとき、このスクリプトは、NIS+ ディレクトリー (**groups\_dir** および **org\_dir** を含む) と、*Domain* に指定されたドメインのシステム・テーブル・オブジェクトを作成します。ただし、**nisserver** はテーブルにデータを転送しません。テーブルに転送する場合は、**nispopulate** を使用します。

ルート・マスター・サーバーをセットアップする場合は、**-r** フラグを使用します。このフラグを使用するには、**nisserver** を実行しているサーバー上でスーパーユーザーでなければなりません。指定されたドメインの非ルート・マスター・サーバーをセットアップするには、**-M** フラグを使用します。このフラグを使用するには、NIS+ マシン上の NIS+ プリンシパルでなければならず、*Domain* の親ディレクトリーに対する書き込み許可を保有する必要があります。新しい非ルート・マスター・サーバーは、既に **rpc.nisd** デーモンを実行している NIS+ クライアントになっていなければなりません (**nisclient** コマンドを参照)。ル

ートと非ルートの両方のドメインのレプリカ・サーバーをセットアップする場合は、**-R** フラグを使用します。このフラグを使用するには、NIS+ サーバー上の NIS+ プリンシパルでなければならず、複製されるドメインの親ディレクトリーに対する書き込み許可を保有している必要があります。

## フラグ

<b>-d</b> <i>Domain</i>	NIS+ ドメインを指定します。デフォルトはそのローカル・ドメインです。
<b>-f</b>	確認のプロンプトを出さずに NIS+ サーバーのセットアップを強制します。
<b>-g</b> <i>GroupName</i>	新しいドメインの NIS+ グループを指定します。 <b>-g</b> フラグは <b>-R</b> フラグと一緒に使用すると無効になります。デフォルト・グループは <b>admin</b> です。
<b>-h</b> <i>HostName</i>	NIS+ サーバーのホスト名を指定します。このサーバーは、ローカル・ドメイン内で有効なホストでなければなりません。ローカル・ドメイン外のホストを指定する場合は、完全修飾ホスト名を使用します。 <b>-h</b> フラグは、非ルート・マスターまたはレプリカ・サーバーをセットアップする場合にのみ有効です。マスター・サーバーのデフォルトは、親ドメインのサーバー・リストと同じリストを使用することです。レプリカ・サーバーのデフォルトは、ローカル・ホスト名を使用することです。
<b>-l</b> <i>Password</i>	ルート・マスター・サーバーのクレデンシャルを作成するためのネットワーク・パスワードを指定します。 <b>-l</b> フラグは、 <b>-r</b> フラグを指定したときにのみ有効です。このフラグを指定しないと、 <b>nissserver</b> スクリプトはログイン・パスワードを入力するためのプロンプトを出しません。
<b>-M</b>	指定されたホストをマスター・サーバーとしてセットアップします。 <b>-M</b> フラグを指定して <b>nissserver</b> コマンドを実行する前に、そのホストで <b>rpc.nisd</b> デーモンが実行されていなければなりません。
<b>-R</b>	指定されたホストをレプリカ・サーバーとしてセットアップします。 <b>-M</b> フラグを指定して <b>nissserver</b> コマンドを実行する前に、そのホストで <b>rpc.nisd</b> デーモンが実行されていなければなりません。
<b>-r</b>	サーバーをルート・マスター・サーバーとしてセットアップします。
<b>-v</b>	詳細モードでスクリプトを実行します。
<b>-x</b>	エコ・モードをオンにします。
<b>-Y</b>	NIS+ サーバーを、NIS 互換モードを指定してセットアップします。デフォルトは、NIS 互換モードを指定しないことです。

## 例

- ドメイン `abc.com.` のルート・マスター・サーバーをセットアップするには、次のように入力します。  
`/usr/lib/nis/nissserver -r -d abc.com.`
- ホスト `abc replica` 上のドメイン `abc.com.` のレプリカ・サーバーをセットアップするには、次のように入力します。  
`/usr/lib/nis/nissserver -R -d abc.com.`  
`/usr/lib/nis/nissserver -R -d abc.com. -h abc replica`
- NIS+ グループ名を `admin-mgr.abc.xyz.com.` と指定して、ホスト `defhost` 上のドメイン `abc.xyz.com.` の非ルート・マスター・サーバーをセットアップするには、次のように入力します。  
`/usr/lib/nis/nissserver -M -d abc.xyz.com.`  
`/usr/lib/nis/nissserver -M -d abc.xyz.com. -h defhost -g admin-mgr.abc.xyz.com.`
- `defhost` 上のドメイン `abc.xyz.com.` の非ルート・レプリカ・サーバーをセットアップするには、次のように入力します。  
`/usr/lib/nis/nissserver -R -d abc.xyz.com. -h defhost`

注: 最後の 3 つの例ではそれぞれ、ホストが、コマンド・ストリングを実行する前に **rpc.nisd** デーモンを実行している NIS+ クライアントでなければなりません。

## 関連情報

**nisaddcred** コマンド、**nisclient** コマンド、**nisgrpadm** コマンド、**nisinit** コマンド、**nismkdir** コマンド、**nispopulate** コマンド、**nissetup** コマンド、および **rpc.nisd** デーモン。

---

## nissetup コマンド

### 目的

NIS+ ドメインを初期化します。

### 構文

```
/usr/lib/nis/nissetup [ -Y ] NIS+Domain
```

### 説明

**nissetup** コマンドは、クライアントにサービスを提供し、システム管理情報を格納するためのドメインを初期化します。**nissetup** は、システム管理情報をドメイン *NIS+Domain* 内に格納する必要があるクライアントにサービスを提供するための、NIS+ ドメインを確立するシェル・スクリプトです。このドメインは、**nissetup** を実行する前にあらかじめ存在しているものでなければなりません (ドメインの詳細な作成方法については、**nismkdir** および **nisinit** を参照してください)。

NIS+ ドメインは、NIS+ ディレクトリーとそのサブディレクトリー **org\_dir** および **groups\_dir** から構成されます。**org\_dir** サブディレクトリーは、システム管理情報を格納し、**groups\_dir** はグループ・アクセス制御に関する情報を格納します。

**nissetup** は、サブディレクトリー **org\_dir** および **groups\_dir** を *NIS+Domain* 内に作成します。**org\_dir** と **groups\_dir** は両方とも、親ドメインのサーバー上で複製されます。これらのサブディレクトリーが作成された後、**nissetup** は、NIS+ がサービスを行う、以下のデフォルト・テーブルを作成します。

- **auto\_master**
- **auto\_home**
- **bootparams**
- **cred**
- **ethers**
- **group**
- **hosts**
- **mail\_aliases**
- **netmasks**
- **networks**
- **passwd**
- **protocols**
- **rpc**

- **services**
- **timezone**

**nissetup** スクリプトは、**nistbladm** コマンドを使用して上記のテーブルを作成します。このスクリプトは簡単にカスタマイズして、セットアップ時にサイト特有のテーブルを追加できます。

注: **nissetup** はデフォルト・テーブルを作成しますが、データを含めたそれらの初期化は行いません。初期化を行うには、**nisaddent** コマンドを使用します。

通常、**nissetup** コマンドは、ドメインごとに 1 回だけ実行されます。

## フラグ

- Y** そのドメインが NIS+ ドメインと NIS ドメインの両方の働きをするように指定します。-**Y** フラグは、認証されていないクライアントでもすべてのシステム・テーブルを読み取れるようにするので、結果的にそのドメインの安全性が低くなります。

## 関連情報

**nisaddent** コマンド、**nisinit** コマンド、**nismkdir** コマンド、および **nistbladm** コマンド。

---

## nisshowcache コマンド

### 目的

共有キャッシュ・ファイルの内容を出力します。

### 構文

```
/usr/lib/nis/nisshowcache [ -v ]
```

### 説明

**nisshowcache** コマンドは、そのサーバー上の NIS+ にアクセスしているすべてのプロセスが共用する、サーバーごとの NIS+ ディレクトリー・キャッシュの内容を出力します。デフォルトでは、

**nisshowcache** は、キャッシュ・ヘッダーと一緒にキャッシュ内のディレクトリー名だけを出力します。共用されるキャッシュは、**nis\_cachemgr** コマンドによって維持されます。

### フラグ

- v** 詳細モードで **nisshowcache** コマンドを設定します。-**v** フラグを指定すると、**nisshowcache** は、各ディレクトリー・オブジェクトの内容 (サーバー名やその汎用アドレスに関する情報など) を出力します。

### ファイル

**/var/nis/NIS\_SHARED\_DIRCACHE** は **nisshowcache** コマンドを含みます。

## 関連情報

**nis\_cachemgr** コマンドおよび **syslogd** デーモン。

---

## nisstat コマンド

### 目的

NIS+ サーバー統計情報を報告します。

### 構文

```
/usr/lib/nis/nisstat [ -H HostName ] [ DirName ]
```

### 説明

**nisstat** コマンドは、その操作に関する統計情報について NIS+ に照会します。これらの統計情報は、リリースごとに異なり、インプリメンテーション間でも異なります。すべてのサーバーからすべての統計情報が入手できるわけではありません。統計情報をサポートしないサーバーに統計情報を要求すると、**nisstat** は単純に **unknown statistic** を戻してきます。

デフォルトでは、統計情報は、デフォルト・ドメインの NIS+ ディレクトリーのサーバーから検索されます。*DirName* にディレクトリーが指定されている場合、そのディレクトリーのサーバーが照会されます。

特定の統計情報を検索するには、次のキーワードのいずれかを使用します。

<b>root server</b>	そのサーバーがルート・サーバーかどうかを報告します。
<b>NIS compat mode</b>	そのサーバーが NIS 互換モードで実行しているかどうかを報告します。
<b>DNS forwarding in NIS mode</b>	NIS 互換モードのサーバーが、DNS にホスト・ルックアップ・コールを転送するかどうかを報告します。
<b>security level</b>	デフォルト・サーバーか、 <i>HostName</i> に指定されたサーバーのセキュリティー・レベルを報告します。
<b>serves directories</b>	デフォルト・サーバーか、 <i>HostName</i> に指定されたサーバーによってサービスを提供されるディレクトリーをリストします。
<b>Operations</b>	次の形式で結果を戻します。  <b>OP=<i>opname</i>:C=<i>calls</i>:E=<i>errors</i>:T=<i>micros</i></b>  <i>opname</i> RPC プロシージャまたは操作を示します。  <i>calls</i> サーバーが実行を開始してから行われた、RPC プロシージャに対する呼び出しの回数を示します。  <i>errors</i> 呼び出しが処理されている間に発生したエラーの数を示します。  <i>micros</i> 最後に行われた 16 回の呼び出しを完了する際の平均時間 (マイクロ秒単位) を示します。
<b>Directory Cache</b>	内部ディレクトリー・オブジェクト・キャッシュに対する呼び出しの回数、そのキャッシュでのヒットの数、ミスの数、ヒット率のパーセンテージを報告します。
<b>Group Cache</b>	内部 NIS+ グループ・オブジェクト・キャッシュに対する呼び出しの回数、そのキャッシュでのヒットの数、ミスの数、およびヒット率のパーセンテージを報告します。
<b>Static Storage</b>	サーバーが、その静的ストレージ・バッファー用に割り当てたバイト数を報告します。
<b>Dynamic Storage</b>	サーバー・プロセスで現在使用されているヒープの量を報告します。
<b>Uptime</b>	サービスが実行している時間を報告します。

## フラグ

**-H** *HostName* *HostName* に指定されたサーバーだけが、**nisstat** コマンドによって照会されます。デフォルトでは、そのディレクトリーのすべてのサーバーが照会されます。*HostName* がディレクトリーにサービスを提供しない場合、統計情報は戻されません。

## 環境

**NIS\_PATH** NIS+ 名が完全修飾でない場合、そのディレクトリーが見つかるまで、指定された各 NIS+ ディレクトリーが検索されます。

## 関連情報

**nisdefaults** コマンド。

---

## nistbladm コマンド

### 目的

NIS+ テーブルを管理します。

### 構文

#### テーブル・エントリーを追加または上書きする

```
nistbladm -a | -A [ -D Defaults ] { Col_name=Value... Tbl_name }
```

```
nistbladm -a | -A [ -D Defaults ] { Entry_Name }
```

注: *Entry\_Name* の構文は [column=value],table です。

#### NIS+ テーブルを作成する

```
nistbladm -c [ -D Defaults ] [ -p Path ] [ -s Sep ] Type Col_name=[ S ][ I ][ C ][ B ][ X ] [ Access ]... Tbl_name
```

注: *Col\_name* の後のフラグはコンマで区切る必要があります。

例:

```
nistbladm -c hobby_tbl name=S,a+r,o+m hobby=S,a+r hobbies.abc.com.
```

#### NIS+ テーブル全体を削除する

```
nistbladm -d Tbl_name
```

#### テーブル・エントリーを編集する

```
nistbladm -m | -E Col_name=Value... Entry_name
```

#### テーブル・エントリーを除去する

```
nistbladm -r | -R { [ Col_name=Value... ] Tbl_name }
```

```
nistbladm -r | -R { Entry_name }
```

## テーブルの属性を更新する

```
nistbladm -u [ -p Path ] [ -s Sep ] [ -t Type ] [ Col_name=Access... ] Tbl_name
```

### 説明

**nistbladm** コマンドは、NIS+ テーブルを管理するために使用します。このコマンドには、テーブルの作成、テーブルの削除、テーブル・エントリーの追加、テーブル・エントリーの修正、テーブル・エントリーの除去という 5 つの基本操作があります。

NIS+ はテーブルまたはエントリーのサイズは制限しませんが、データのサイズは、NIS+ サーバーのパフォーマンスとディスク・スペース要件に影響を与えます。NIS+ は、大量のデータ (ファイルなど) を格納するように設計されていません。その代わりに他のサーバー上にあるファイルを指すポインタを保管してください。NIS+ は、合計 10M バイトまで、最高 10,000 個のオブジェクトをサポートできます。より多くのストレージ・スペースが必要な場合、ドメインの階層を作成するか、または NIS+ 内に実際のデータを保管するのではなく、テーブルに保管されたデータを、実際のデータを指すポインタとして使用します。

テーブルを作成するには、そのディレクトリーがあらかじめ存在している必要があります、そのディレクトリーに対する作成権がなければなりません。必ずテーブル名、テーブル・タイプ、および列定義のリストを指定してください。Type は、NIS+ が、エントリーが正しいタイプのものであるかどうかを検査する場合に基準として使用する文字列です。

テーブルを削除するには、テーブルが格納されているディレクトリーに対する破棄権を保有している必要があります。エントリーを修正 (追加、変更、または削除) するには、テーブルまたは個々のエントリーに対する修正権が必要です。

### フラグ

**-a** NIS+ テーブルに新しいエントリーを追加します。コマンド行に *Col\_name=Value* の組を指定することによって、エントリーの内容を作成します。

#### 注:

1. NIS+ テーブルにエントリーを追加するときは、各カラムに値を指定する必要があります。
2. 値の文字列を入力するときは、終了文字を単一引用符 (') または二重引用符 (") で囲みます。これらの文字は、等号 (=)、コンマ (,)、左大括弧 ((), 右大括弧 ()), およびスペース ( ) です。これらは索引付きの名前の中で NIS+ によって散らされて用います。

**-a** フラグを指定すると、**nistbladm** コマンドは、目的のカラム内にある既存の値を上書きするエントリーを追加しようとする、エラーを報告します。**nistbladm** コマンドは既存のエントリー値を自動的に上書きしません。(エントリーの上書きについては、**-A** フラグを参照してください。)

**-A** **nistbladm** コマンドに、既存のエントリー値を上書きするように強制します。*Col\_name* に既に値が含まれている場合でも、**nistbladm** は古い値を新しい値によって上書きします。

**-a** フラグとは異なり、**nistbladm** コマンドはエラーを戻しません。



**-c** *Tbl\_name*

パラメーター *Tbl\_name* 内で指名された新しい NIS+ テーブルを作成します。テーブルを作成するときは、必ず、テーブル・タイプ、エントリーのタイプ、および列定義のリストを指定してください。列定義の構文は、*Col\_name*=[ *Flags* ] [ *Access* ] です。パラメーター *Flags* は、次の値を持つことができます。

- S** カラムの値で検索が実行できることを指定します。
- I** 検索でカラム値の大文字小文字を無視することを指定します。このフラグは、**S** フラグと組み合わせて使用する場合にのみ有効です。
- C** カラムの値を暗号化します。
- B** カラムの値をバイナリー・データとして設定します。**B** フラグが設定されていない場合、カラム値は null 終了 ASCII 文字列です。このフラグは、**S** フラグと組み合わせて使用する場合にのみ有効です。
- X** カラムの値を XDR エンコードされたデータとして設定します。**X** フラグは、**B** フラグと組み合わせて使用する場合にのみ有効です。

新しく作成されたテーブルには必ず最低でも 1 つのカラムが含まれていなければならない、そこには検索可能カラムが少なくとも 1 つは含まれている必要があります。つまり、*Tbl\_name* にカラムが 1 つしかない場合、そのカラムは検索可能でなければならないということです。

**-d** *Tbl\_name*

パラメーター *Tbl\_name* で示されたテーブル全体を削除します。このテーブルは削除前に空にしておく必要があります。(テーブルの内容を削除するには、**-R** フラグを使用します。)

**-D**

新しいオブジェクトの作成時に使用するデフォルト・セットを指定します。defaults 文字列は、コロンで区切られた一連のトークンです。これらのトークンは、汎用オブジェクト属性に使用するデフォルト値を表します。

**ttl=Time**

**nistbladm** コマンドによって作成されたオブジェクトのデフォルトの存続時間を設定します。値 *Time* は、**nischttl** コマンドによって定義されたフォーマットで指定する必要があります。デフォルト値は 12 時間です。

**owner=Ownername**

NIS+ プリンシパル *Ownername* が、作成されたオブジェクトを所有するように指定します。デフォルト値は、オブジェクトを作成するために **nistbladm** コマンドを実行するプリンシパルと同じです。

**group=Groupname**

グループ *Groupname* が、作成されたオブジェクトのグループ・オーナーとなるように指定します。デフォルト値は NULL です。

**access=Rights**

所定のオブジェクトに関して認可される 1 組のアクセス権限を指定します。値 *Rights* は、**nischmod** コマンドによって定義されたフォーマットで指定する必要があります。デフォルト値は `--rmcdr--r--` です。

**-e** *Entry\_name*

*Entry\_name* によって指定されたエントリーを編集します。*Entry\_name* は、1 つのエントリーだけを一意的に表していなければなりません。*Entry\_name* の値を編集するときには、そのエントリーの索引付きの名前を変更することもできます。

**注:** そのエントリーの新しい索引付き名 (編集の結果得られる名前) が別のエントリーの名前と一致する場合、**nistbladm** コマンドは失敗し、エラー・メッセージを戻します。

**-E** *Entry\_name*

*Entry\_name* によって指定されたエントリーを編集します。*Entry\_name* は、1 つのエントリーだけを一意的に表していなければなりません。

**注:** 新しい索引付きの名前が別のエントリーの名前と一致する場合、**-E** フラグが自動的に、その既存エントリーを、編集して得られたエントリーで上書きします。そのため、結果的には 2 つのエントリーが 1 つのエントリーに置き換えられることとなります。

<b>-m</b>	<b>-E</b> と同じ機能です。
<b>-r</b>	テーブルからエントリーを除去します。 <i>Entry_value</i> 内にあるその索引付きの名前で該当のエントリーを表すか、コマンド行の一連の <i>Col_name=Value</i> の組によってそのエントリーを表します。 <b>-r</b> フラグを指定すると、 <b>nistbladm</b> コマンドは、索引付きの名前または <i>column=value</i> の組が複数のエントリーと一致する場合は失敗になります。
<b>-R</b>	テーブルから複数のエントリーを除去します。 <b>-R</b> フラグは、 <b>nistbladm</b> コマンドに、除去の基準を満たすエントリーを全部除去するように強制します。その基準が <i>null</i> の場合 ( <i>column=value</i> の組も索引付きの名前も指定しない場合)、そのテーブルから <b>すべての</b> エントリーが除去されます。
<b>-u</b>	テーブルの属性を更新します。これによって、テーブルの連結パス、分離文字、カラム・アクセス権限、およびテーブル・タイプ文字列が変更できます。このフラグでは、カラム数も検索可能カラム数も変更できません。
<b>-p Path</b>	テーブルの作成または変更時にテーブルの検索パスを指定します。 <b>nis_list</b> 機能を起動すると、フラグ <b>FOLLOW_PATH</b> を指定して、クライアント・ライブラリーに対し、検索基準を満たすエントリーがない場合、 <i>Path</i> 内のテーブルの検索を継続するように指示できます。このパスは、コロンで区切られたテーブル名の番号付きリストを含んでいます。パス内の名前は完全修飾でなければなりません。
<b>-s Sep</b>	テーブルの作成または修正時にテーブルのセパレーター文字を指定します。セパレーター文字は、 <b>niscat</b> コマンドが標準出力にテーブルを書き出すときに使用します。セパレーター文字の目的は、テーブルが ASCII 形式のときにカラム・データを区切ることです。デフォルト値は <i>&lt;スペース&gt;</i> です。
<b>-t Type</b>	テーブルを修正するときは、そのテーブルの <i>Type</i> 文字列を指定します。

## 終了状況

- 0 正常終了。
- 1 エラーが発生しました。

## 環境変数

<b>NIS_DEFAULTS</b>	NIS+ 標準デフォルトをオーバーライドするデフォルト文字列が入っています。ただし、 <b>-D</b> フラグ値によって別の値を指定すると、それらの値が <b>NIS_DEFAULTS</b> 変数と標準デフォルトの両方をオーバーライドします。
<b>NIS_PATH</b>	<i>Tbl_name</i> が完全修飾でない場合、この変数を設定すると、 <b>nistbladm</b> に対し、そのテーブルが見つかるまで指定された各ディレクトリーを検索するように指示します。

## 例

- ディレクトリー *abc.com*. 内に、2 つの検索可能カラム *name* と *hobby* を含む、*hobbies* という名前のタイプ *hobby\_tbl* のテーブルを作成するには、次のように入力します。

```
nistbladm -c hobby_tbl name=S,a+r,o+m hobby=S,a+r hobbies.abc.com.
```

カラム *name* は、すべて (オーナー、グループ、およびワールド) についての読み取りアクセスと、オーナーのみについての修正アクセスを保有します。カラム *hobby* は、すべてについての読み取りアクセスを保有していますが、だれからも修正できません。

アクセス権限が指定されていない場合、テーブルのアクセス権限は、標準デフォルトか、**NIS\_DEFAULTS** 変数によって指定されたデフォルトになります。

- hobbies* テーブルにエントリーを追加するには、次のように入力します。

```
nistbladm -a name=bob hobby=skiing hobbies.abc.com.
```

```
nistbladm -a name=sue hobby=skiing hobbies.abc.com.
```

```
nistbladm -a name=ted hobby=swimming hobbies.abc.com.
```

3. 連結パスを追加するには、次のように入力します。

```
nistbladm -u -p hobbies.xyz.com.:hobbies.def.com. hobbies
```

4. テーブルからスキーの愛好者を削除するには、次のように入力します。

```
nistbladm -R hobby=skiing hobbies.abc.com.
```

**注:** この例で **-r** フラグを使用すると、2 つのエントリーに値 **skiing** が含まれているため失敗します。

5. フラグを設定せずに名前が指定されているカラムを含むテーブルを作成するには、次のように入力します。

```
nistbladm -c notes_tbl_name=S,a+r,o+m note=notes.abc.com.
```

このコマンド・ストリングは、2 つのカラム **name** と **note** を含むタイプ **notes\_tbl** のテーブル **notes.abc.com.** を作成します。**note** カラムは検索可能ではありません。

## 関連情報

**niscat** コマンド、**nischmod** コマンド、**nischown** コマンド、**nisdefaults** コマンド、**nismatch** コマンド、および **nissetup** コマンド。

---

## nistest コマンド

### 目的

条件式を使用して、NIS+ ネーム・スペースの状態を戻します。

### 構文

```
nistest [ [ -A ] [ -L ] [ -M ] [ -P ] ] [ -a | -t Type ] Object
```

```
nistest [ -A ] [ -L ] [ -M ] [ -P ] [ -a Rights ] IndexedName
```

### 説明

**nistest** コマンドは、シェル・スクリプトなどのプログラムが、オブジェクトおよびエントリーの有無、タイプ、アクセス権限についてテストするための方法を提供します。エントリーは索引付きの名前を使用して指定します (**nismatch** コマンドを参照)。

### フラグ

- A** そのテーブル内の全データと、戻される初期テーブルの連結パス内にあるテーブルの全データを指定します。このフラグは、索引付きの名前を使用するとき、またはリンクの後に続くときにのみ有効です。
- L** リンクの後に続きます。 *Object* によって指定されたオブジェクトか、*IndexedName* のテーブル名コンポーネントが **LINK** タイプのオブジェクトを指定している場合、このスイッチが存在するときはこのリンクの後に続きます。

<b>-M</b>	指名されたデータのマスター・サーバーにだけルックアップを送信するように指定します。これによって、最新の情報を表示することが保証されますが、場合によってはマスター・サーバーが使用中であるという欠点があります。
<b>-P</b>	初期検索が失敗であった場合、ルックアップをテーブルの連結パスの後に続けるように指定します。このフラグは、索引付きの名前を使用するとき、またはリンクの後に続くときにのみ有効です。
<b>-a Rights</b>	現行プロセスに、指定されたオブジェクトまたはエントリーについて、必要かつ望まれているアクセス権限があるか確認します。このアクセス権限は、 <b>nischmod</b> コマンドの場合と同じ方法で指定します。
<b>-t Type</b>	<i>Object</i> のタイプをテストします。 <i>type</i> の値には、次のいずれかを指定できます。
<b>G</b>	オブジェクトがグループ・オブジェクトの場合、真を返します。
<b>D</b>	オブジェクトがディレクトリー・オブジェクトの場合、真を返します。
<b>T</b>	オブジェクトがテーブル・オブジェクトの場合、真を返します。
<b>L</b>	オブジェクトがリンク・オブジェクトの場合、真を返します。
<b>P</b>	オブジェクトが専用オブジェクトの場合、真を返します。

## リターン値

<b>0</b>	正常終了。
<b>1</b>	オブジェクトが、存在しない、指定されたタイプでない、またはそのアクセスがない (あるいはこれら全部の) ために失敗しました。
<b>2</b>	使用方法が正しくないために失敗しました。

## 例

1. アクセス権限についてテストするときは、現行ユーザーに指定された権限が認可されている場合、**nistest** は正常終了 (0) を返します。したがって、アクセス権限をテストするには次のように入力します。

```
nistest \-a w=mr skippy.domain
```

認証された NIS+ クライアントがすべて、**skippy.domain** で指名されたオブジェクトに対する読み取りアクセスと修正アクセスを保有しているかテストします。

2. テーブル内の特定のエントリーに関するアクセスについてのテストは、索引付きの名前の構文を使用して実行できます。次の例は、パスワード・テーブル内のエントリーが修正可能かどうかを確認するためのテストをします。

```
nistest \-a o=m '[uid=99],passwd.org_dir'
```

## 環境

### NIS\_PATH

この変数が設定されていて、NIS+ 名が完全修飾でない場合、指定された各ディレクトリーが、該当のオブジェクトが見つかるまで検索されます (**nisdefaults** を参照)。

## 関連情報

**nischmod** コマンドおよび **nisdefaults** コマンド。

---

## nistoldif コマンド

### 目的

ユーザー、グループ、ネーム・レゾリューション、および rpc データを rfc 2307 準拠形式にエクスポートします。

### 構文

```
nistoldif -d Suffix [ -a BindDN -h Host -p Password [-n Port ] ] [ -f Directory ] [ -y domain ] [ -S Schema ] [ -k KeyPath -w SSLPassword ] [ -s Maps ] [ -m ldap_mapname ]
```

### 説明

**nistoldif** コマンドは、**passwd**、**group**、**hosts**、**services**、**protocols**、**rpc**、**networks**、**netgroup**、および **automount** からのデータを、rfc2307 に準拠した形式に変換します。これは最初に NIS からデータを読み取ることを試み、NIS マップが見つからない場合は、フラット・ファイルにフォールバックします。

コマンド行にサーバー情報 (**-a**、**-h**、および **-p** フラグ) が与えられている場合は、データはサーバーに直接書き込まれます。データが、サーバー上の既存のエントリーと矛盾する (エントリーが既に存在するため、または **uid** または **gid** が既に存在するため) 場合は、警告が印刷されます。サーバー情報が与えられていない場合、データは LDIF 内の **stdout** に書き込まれます。いずれの場合も、**nistoldif** は接尾部用のエントリーは追加しません。したがって、そのエントリーが存在しない場合、サーバーにデータを追加する試みは失敗します。このエントリーはサーバーのセットアップ中に、通常 **mksecldap** コマンドにより、追加されます。

変換は正確ではありません。rfc2307 定義の制限により、いくつかの属性は大/小文字を区別しないで定義されます。例えば、TCP、Tcp、および tcp はすべて LDAP サーバーにとっては同じプロトコル名です。2^31-1 より大きい Uid および gid は、ストレージに相当する負の 2 の補数に変換されます。

**nistoldif** コマンドは **/etc/security/ldap/sectoldif.cfg** ファイルを読み、**passwd**、**group**、**hosts**、**services**、**protocols**、**rpc**、**networks** および **netgroup** データがエクスポートされる先のサブツリーにどのような名前を付けるかを決定します。このファイルで指定される名前は、**-d** フラグで指定される基本 DN の下にサブツリーを作成するために使用されます。詳しくは、**/etc/security/ldap/sectoldif.cfg** ファイルの資料を参照してください。

### フラグ

- a** LDAP サーバーへの接続に使用される管理バインド DN を指定します。このフラグを使用する場合は **-h** と **-p** も使用する必要があります、データは LDAP サーバーに直接書き込まれます。
- d** データをその下に追加すべき接尾部を指定します。
- f** フラット・ファイルを探すディレクトリー、または自動マウント・マップ・ファイルの名前を指定します。このフラグを使用しない場合、**nistoldif** は **/etc** 内でファイルを探します。このフラグが必要になるのは、自動マウント・マップに対してです。
- h** LDAP サーバーを実行するホスト名を指定します。このフラグを使用する場合は、**-a** と **-p** も使用する必要があります、データは LDAP サーバーに直接書き込まれます。このフラグは、自動マウント・データの場合は無視されます。
- k** SSL キー・パスを指定します。このフラグを使用する場合は **-w** も使用する必要があります。
- m** LDAP サーバーの自動マウント・マップを指定します。

- n** LDAP サーバーに接続するポートを指定します。このフラグを使用する場合は、**-a**、**-h**、および **-p** も使用する必要があります。使用しないと、デフォルトの LDAP ポートが使用されます。
- p** LDAP サーバーへの接続に使用するパスワードを指定します。このフラグを使用する場合は、**-a** と **-h** も使用する必要があります。データは LDAP サーバーに直接書き込まれます。
- s** サーバーに書き込まれるマップのセットを指定します。このフラグの後には、マイグレーションすべきマップを表す文字のリストを続ける必要があります。このフラグを使用しない場合は、すべてのマップがマイグレーションされます。文字は、automount は **a**、netgroup は **e**、group は **g**、host は **h**、network は **n**、protocol は **p**、rpc は **r**、service は **s** および passwd は **u** です。
- S** ユーザーおよびグループに使用する LDAP スキーマを指定します。これは RFC2307 または RFC2307AIX のいずれかを指定できます。RFC2307AIX は拡張 AIX スキーマのサポートを提供します。このフラグを使用しない場合は、RFC2307 がデフォルトです。
- w** SSL パスワードを指定します。このフラグを使用する場合は **-k** も使用する必要があります。
- y** マップを読み取る NIS ドメインを指定します。このフラグを指定しないと、デフォルトのドメインが使用されます。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0** エラーはありません。マップが見つからないことはエラーと見なされないことに注意してください。
- >0** エラーが発生しました。

## セキュリティ

アクセス制御: root ユーザーのみがこのコマンドを実行できます。

## 例

1. NIS マップを、ドメイン **austin.ibm.com** (**/tmp/etc** 内のフラット・ファイルにフォールバック) から LDIF の接尾部 **cn=aixdata** の下にエクスポートするには、次のように入力します。

```
nistoldif -d cn=aixdata -y austin.ibm.com -f /tmp/etc > ldif.out
```

2. ホストおよびサービス・マップを、デフォルト・ドメイン (**/etc** 内のフラット・ファイルにフォールバック) から LDAP サーバー **ldap.austin.ibm.com** に、管理者バインド DN **cn=root**、パスワード **secret** を使用し、接尾部 **cn=aixdata** の下にエクスポートするには、次のように入力します。

```
nistoldif -d cn=aixdata -h ldap.austin.ibm.com -a cn=root -p secret -s hs
```

3. **/etc/auto\_master** 自動マウント・マップ・ファイルを LDIF に変換するには、次のように入力します。

```
nistoldif -s a -f /etc/auto_master > ldif.out
```

4. 自動マウント・データを除去するには、LDIF ファイルを手動で作成する必要があります。例えば、**dc=austin,dc=ibm,dc=com** サフィックスの **auto\_home** 自動マウント・マップにユーザー **user1** が誤って追加されたため、削除する必要があるとします。この場合、次のような LDIF を作成します。

```
# cat /tmp/del_user1.ldif
dn: automountKey=user1,automountMapName=auto_home,dc=austin,dc=ibm,dc=com
changetype: delete
```

その後、次のコマンドを実行します。

```
ldapmodify -f /tmp/del_user1.ldif
```



5. 自動マウント・データを編集するには、LDIF ファイルを手動で作成する必要があります。例えば、`dc=austin,dc=ibm,dc=com` サフィックスの `auto_home` 自動マウント・マップでユーザー `user2` に誤ったマウント・ポイントが指定されたため、`/home/user2` の正しい場所に変更する必要があるとします。この場合、次のような LDIF を作成します。

```
# cat /tmp/ch_user2.ldif
dn: automountKey=user2,automountMapName=auto_home,dc=austin,dc=ibm,dc=com
changetype: modify
replace: automountInformation
automountInformation: /home/user2
```

その後に、次のコマンドを実行します。

```
ldapmodify -f /tmp/ch_user2.ldif
```

## ファイル

`/usr/sbin/nistoldif` `nistoldif` コマンドが入っています。

## 関連情報

`mksecdap` コマンド。

`/etc/security/ldap/sectoldif.cfg` ファイル。

---

## nisupdkeys コマンド

### 目的

NIS ディレクトリー・オブジェクトの公開鍵を更新します。

### 構文

```
/usr/lib/nis/nisupdkeys [ -a ] | [ -C ] [ -H Hostname ] [ -s ] [ Dirname ]
```

### 説明

`nisupdkeys` コマンドは、NIS+ ディレクトリー・オブジェクトの公開鍵を更新します。NIS+ サーバーの公開鍵を変更するときは、新しい鍵を、そのサーバーを参照するすべてのディレクトリー・オブジェクトに伝搬する必要があります。`nisupdkeys` は、ディレクトリー・オブジェクトを読み取り、そのディレクトリーの各サーバーの公開鍵をコピーしようとします。公開鍵がディレクトリー・オブジェクト内に置かれると、新しい鍵を反映するようにオブジェクトが修正されます。

`Dirname` が存在する場合、そのディレクトリー・オブジェクトが更新されます。存在しない場合、デフォルト・ドメインのディレクトリー・オブジェクトが更新されます。`nisupdkeys -s` は、`Hostname` によってサービスを提供される全ディレクトリーのリストを入手し、呼び出し元が必要なアクセス権を持っていることを想定して、それらのディレクトリー・オブジェクトを更新します。ディレクトリーのリストは、`nisstat` コマンドによって入手することもできます。

`nisupdkeys` を実行するまでに、新しいアドレス/公開鍵を確実にすべてのレプリカ・サーバーに伝搬しておく必要があります。



## フラグ

- a** ディレクトリー・オブジェクト内の NIS+ サーバーの汎用アドレスを更新します。 **-a** フラグは、TCP/IP トランスポート・ファミリーの場合にしか使用できません。このフラグは、サーバーの IP アドレスが変更されたときに使用するようになっています。新しいアドレスは、そのサーバー上の **gethostname** を使用して解決されます。このレゾリューションを正常に行うには、**/etc/nsswitch.conf** ファイルがそのサーバーのエントリーの正しいソースを指していないければなりません。
- C** 公開鍵をクリアします。公開鍵のないサーバーとの通信には、セキュア・リモート・プロシージャ・コールを使用する必要がありません。
- H Hostname** 現在のドメインのディレクトリー・オブジェクトの中で、*Hostname* という名前のサーバーの鍵を更新します。このホスト名が完全修飾でない場合、**nisupdkeys** は、そのサーバーがデフォルト・ドメイン内にあると想定します。 *Hostname* がディレクトリーにサービスを提供しない場合、何も行われません。
- s** 必要なアクセス権があることを前提として、サーバー *Hostname* によってサービスが提供されるすべての NIS+ ディレクトリー・オブジェクトを更新します。ディレクトリー・オブジェクトを更新する許可がない場合、その更新は失敗し、その旨が通知されます。 *Hostname* 上の **rpc.nisd** が、そのサービスの対象であるサーバーのリストを戻すことができない場合、**nisupdkeys** はエラー・メッセージを戻します。このエラー・メッセージが出されたら、**nisupdkeys** を何度か (**rpc.nisd** がサービスを提供する NIS+ ディレクトリーごとに 1 回ずつ) 起動しなければなりません。
- Dirname* ディレクトリー *Dirname* のディレクトリー・オブジェクトの鍵を更新します。

## 例

1. abc.def. ドメインのサーバーの鍵を更新するには、次のように入力します。  
`nisupdkeys abc.def.`
2. abc.def. ドメインでサービスを行うホスト xyzserver の鍵を更新するには、次のように入力します。  
`nisupdkeys -H xyzserver abc.def.`
3. abc.def. ドメイン内のホスト xyzserver の鍵をクリアするには、次のように入力します。  
`nisupdkeys -CH xyzserver abc.def.`
4. xyzserver によってサービスが行われるすべてのディレクトリー・オブジェクト内の鍵を更新するには、次のように入力します。  
`nisupdkeys -sH xyzserver`

## セキュリティ

アクセス制御: **nisupdkeys** コマンドを使用するには、NIS+ ディレクトリー・オブジェクトに対する修正権がなければなりません。

## ファイル

`/usr/lib/nis` **nisupdkeys** コマンドが常駐するディレクトリー。

## 関連情報

**chkey** コマンド、**nisaddcred** コマンド、および **niscat** コマンド。

**gethostbyname** サブルーチン。

---

## nl コマンド

### 目的

ファイル内の行に番号を付けます。

### 構文

```
nl [ -b Type ] [ -f Type ] [ -h Type ] [ -l Number ] [ -d Delimiter ] [ -i Number ] [ -n Format ] [ -v Number ] [ -w Number ] [ -p ] [ -s Separator ] [ File ]
```

### 説明

**nl** コマンドは、*File* パラメーター (デフォルトでは標準入力) を読み取り、入力データ内の各行に番号を付けて、その番号付き行を標準出力に書き出します。出力では、**nl** コマンドはコマンド行で指定されたフラグに従って行の左端に番号を付けます。

入力テキストは論理ページに書き込む必要があります。各論理ページはヘッダー、本文、フッター部から構成されます (いずれかの部分に入力がなくてもかまいません)。 **-p** フラグを使用しない限り、**nl** コマンドは、各論理ページの始めに行番号をリセットします。行番号付けフラグは、ヘッダー、本文、フッター部とは独立して設定できます (例えば、ヘッダーおよびフッター行に番号を付け、テキスト行には付けられないようにできます)。

ファイル内では、次の区切り文字のみが入っている行を使って、論理ページの各部分の先頭を示すシグナルを送ります。

行の内容	先頭
\\:\\:	ヘッダー
\\:	本文
\\:	フッター

コマンド行ではファイルを 1 つしか指定できません。フラグとファイル名はどんな順序でもリストできます。

### フラグ

デフォルトではすべてのパラメーターが設定されます。デフォルトの設定を変更する場合は、次のフラグを使用します。 **-s** フラグを除き、デフォルト値を表示させるには変数を指定しないで **-n** フラグを入力します。

<b>-b</b> <i>Type</i>	番号を付ける本文の行を選択します。 <i>Type</i> 変数に指定できる値は次のとおりです。 <b>a</b> 全ての行に番号を付けます。 <b>t</b> ブランク行や、タブなどの非グラフィック文字を含む行には番号を付けません。(デフォルト) <b>n</b> どの行にも番号を付けません。 <b>p</b> <i>Pattern</i> <i>Pattern</i> 変数で指定された行にだけ番号を付けます。
-----------------------	---

<b>-d</b> <i>Delimiter</i>	論理ページ部の始めの区切り文字として <i>Delimiter</i> 変数で指定した 2 文字を使用します。デフォルトの区切り文字は \: (バックスラッシュ、コロン) です。ASCII 文字 2 文字、1 バイト拡張文字 2 文字、または拡張文字 1 文字を指定できます。 <b>-d</b> フラグのあとに 1 バイト文字を 1 文字入力した場合、2 番目の文字はデフォルト (コロン 1 つ) となります。バックスラッシュを区切り文字として使用する場合はバックスラッシュを 2 つ (\\) 入力します。
<b>-f</b> <i>Type</i>	番号を付ける論理ページのフッターの行を選択します。 <i>Type</i> 変数に使用できる値は、 <b>-b</b> フラグと同じです。 <i>Type</i> 変数のデフォルト値は <b>n</b> です (行番号を付けない)。
<b>-h</b> <i>Type</i>	番号を付ける論理ページのヘッダーの行を選択します。 <i>Type</i> 変数に使用できる値は、 <b>-b</b> フラグと同じです。 <i>Type</i> 変数のデフォルト値は <b>n</b> です (行番号を付けない)。
<b>-i</b> <i>Number</i>	論理ページの行番号を <i>Number</i> 変数に指定した数だけ増分します。 <i>Number</i> 変数のデフォルト値は 1 です。 <i>Number</i> 変数の範囲は、1 から 250 です。
<b>-l</b> <i>Number</i>	(L の小文字) <i>Number</i> パラメーターに指定した値の連続する空白行数を、空白行の 1 行分として数える数として使用します。例えば、 <b>-l3</b> は連続した空白行 3 行を 1 行と数えます。 <i>Number</i> 変数のデフォルト値は 1 です。このフラグが機能するのは、 <b>-ha</b> 、 <b>-ba</b> 、または <b>-fa</b> オプションが設定されている場合です。 <i>Number</i> 変数の範囲は、1 から 250 です。
<b>-n</b> <i>Format</i>	<i>Format</i> に指定した値を行番号付けフォーマットとして使用します。指定できる値は次のとおりです。  <b>ln</b> 左に揃い、先行ゼロが抑制されます。  <b>rn</b> 右に揃い、先行ゼロが抑制されます (デフォルト)。  <b>rz</b> 右に揃い、先行ゼロはそのままになります。
<b>-p</b>	論理ページ区切り文字の位置で番号付けを再開しません。
<b>-s</b> <i>Separator</i>	<i>Separator</i> 変数で指定した文字でテキストと行番号を区切ります。 <i>Separator</i> 変数のデフォルト値はタブ文字です。
<b>-v</b> <i>Number</i>	論理ページの初期行番号を、 <i>Number</i> 変数に指定した値に設定します。 <i>Number</i> 変数のデフォルト値は 1 です。 <i>Number</i> 変数の範囲は、0 から 32767 です。
<b>-w</b> <i>Number</i>	<i>Number</i> 変数に指定した値を行番号の文字数として設定します。 <i>Number</i> 変数のデフォルト値は 6 です。 <i>Number</i> 変数の範囲は 1 から 20 です。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0**      正常終了。
- >0**    エラーが発生しました。

## 例

1. 非空白行にのみ番号を付けるには、次のように入力します。

```
nl chap1
```

ここでは、本文中の非空白行にのみ番号が付いた状態で、chap1 の番号付きリストが表示されます。chap1 に区切り文字 \:|\+: あるいは \: が入っていないと、ファイル全体が本文として見なされません。

2. すべての行に番号を付けるには、次のように入力します。

```
nl -ba chap1
```

ここでは、空白行を含めて本文中のすべての行に番号が付けられます。ほとんどの場合は、このフォーマットの **nl** コマンドで十分です。

3. 異なる行番号フォーマットを指定するには、次のように入力します。

```
nl -i10 -nrz -s:: -v10 -w4 chap1
```

ここでは、chap1 の各行に 10 (-v10) で始まる 10 刻み (-i10) の番号を付けます。先行ゼロ (-nrz) を含めて、各番号を表す 4 桁 (-w4) が表示されます。行番号は 2 つのコロン (-s : :) でテキストから分離されます。

例えば、chap1 に次のテキストが入っている場合、

```
A not-so-important  
note to remember:
```

```
You can't kill time  
without injuring eternity.
```

番号付きリストは次のようになります。

```
0010::A not-so-important  
0020::note to remember
```

```
0030::You can't kill time  
0040::without injuring eternity.
```

空白行には番号が付いていないことに注意してください。これを実行するには、**-ba** フラグを例 2 に示したとおりに使用します。

## ファイル

`/usr/bin/nl` `nl` コマンドを含みます。

## 関連情報

オペレーティング・システムおよびデバイスのマネージのファイルおよび入出力ダイレクトのセクション。

`pr` コマンド。

---

## nlssrc コマンド

### 目的

サブシステムまたはサブシステムのグループの状況を正規形式で入手します。

### 構文

```
nlssrc [-h host] -a
```

```
nlssrc [-h host] -g group_name
```

```
nlssrc [-h host] [-l] [-c] -s subsystem_name
```

```
nlssrc [-h host] [-l] [-c] -p subsystem_pid
```

`nlssrc` の初めの 2 つの使用法の構文は、`lssrc` とまったく同じ出力を生成します。後の 2 つの使用法の構文は、`lssrc` の正規形式で出力を生成します。

## 説明

**nlssrc** コマンドは、サブシステムまたはサブシステムのグループの状況を正規形式で入手するために使用します。**nlssrc -c** コマンドは、**lssrc** コマンドから、サポートされるサブシステムの言語に依存しない出力を得るために使用します。状況は、インストールされた言語のロケールに関係なく、英語で表示されます。**-c** フラグがない場合、**nlssrc** コマンドはデーモンのロケールを使用する **lssrc** コマンドを呼び出します。

## フラグ

- a** 定義されたすべてのサブシステムの現行状況をリストします。
- c** サポートされるサブシステムの正規 **lssrc** 出力を要求します。
- g** *group\_name*  
状況を入手するサブシステムのグループを指定します。 *group\_name* パラメーターがサブシステム・オブジェクト・クラスに含まれていない場合、コマンドは失敗します。
- h** *host*  
この状況アクションの要求先の外部ホストを指定します。ローカル・ユーザーは、 **root** として実行している必要があります。リモート・システムは、リモートのシステム・リソース・コントローラー (SRC) 要求を受け入れるように構成されている必要があります。つまり、**srcmstr** デーモン (**/etc/inittab** を参照) を **-r** フラグを使用して開始し、 **/etc/hosts.equiv** ファイルまたは **.rhosts** ファイルはリモート要求ができるように構成する必要があります。
- l** サブシステムが現行状況を長い形式で送信するように要求します。長い状況は状況の要求をサブシステムに送信することを必要とします。状況を戻すのはサブシステムの責任です。
- p** *subsystem\_pid*  
状況を入手する *subsystem\_pid* パラメーターの特定のインスタンスまたは、状況サブサーバー要求を入手するサブシステムの特定のインスタンスを指定します。
- s** *subsystem\_name*  
状況を入手するサブシステムを指定します。 *subsystem\_name* パラメーターは、サブシステムの実際の名前または同義語名にすることができます。 *subsystem\_name* パラメーターがサブシステム・オブジェクト・クラスに含まれていない場合、コマンドは失敗します。

## セキュリティ

このコマンドを実行するには、 **root** 権限は必要ありません。

## 終了状況

- 0** コマンドが正常に実行されました。
- 1** コマンドが失敗しました。

## 制約事項

このコマンドは、**cthags** および **cthats** サブシステムにのみ適用されます。

## 標準出力

**-h** フラグを指定すると、このコマンドの使用ステートメント (使用法) が標準出力に書き込まれます。

## 標準エラー

エラー・メッセージは、標準エラー（および `ctsnap.host_name.nnnnnnnn.log` ファイル）に書き込まれます。

## 例

1. `ctsubsys` と呼ばれるサブシステムから英語の `nlssrc` 出力を入手するには、次のように入力します。

```
nlssrc -c -ls ctsubsys
```

2. 次の例は、同じ情報を異なるフォーマットで示しています。

```
nlssrc -ls ctsubsys (locale-dependent)
```

```
Subsystem Group PID Status
ctsubsys ctsubsys 6334 active
2 locally-connected clients. Their PIDs:
15614 23248
HA Subsystem domain information:
Domain established by node 5
Number of groups known locally: 1
Number of local
Group Name providers providers/subscribers
ha_filesys 7 1 0
```

```
nlssrc -ls ctsubsys -c (canonical form)
```

```
Number of local clients: 2
PIDs: 15614 23248
HA Subsystem domain information:
Domain established by node 5.
Number of known local groups: 1
Group Name: ha_filesys
Providers: 7
Local Providers: 1
Local Subscribers: 0
```

## 位置

`/usr/sbin/rsct/bin/nlssrc` `nlssrc` コマンドが入っています。

## ファイル

`/tmp/ctsup` 出力ファイルを含むデフォルト・ディレクトリーのロケーション。

`/tmp/ctsup/ctsnap.host_name.nnnnnnnn.log`

コマンド実行のログ・ファイルのロケーション。ここで `nnnnnnnn` はタイム・スタンプであり、`host_name` はコマンドを実行するホストの名前。

`tmp/ctsup/ctsnap.host_name.nnnnnnnn.tar.Z`

収集されたデータを含む圧縮 `tar` ファイルのロケーション。ここで `nnnnnnnn` はタイム・スタンプであり、`host_name` はコマンドを実行するホストの名前。

## 関連情報

コマンド: `lssrc(1)`

---

## nm コマンド

### 目的

オブジェクト・ファイル、実行可能ファイル、およびオブジェクト・ファイル・ライブラリーのシンボルの情報を表示します。

### 構文

```
nm [ -A ] [ -C ] [ -X {32|64|32_64|d64| any} ] [ -f ] [ -h ] [ -l ] [ -p ] [ -r ] [ -T ] [ -v ] [ -B | -P ] [ -e | -g | -u ] [ -d | -o | -x | -t Format ] File ...
```

### 説明

**nm** コマンドは、オブジェクト・ファイル、実行可能ファイル、またはオブジェクト・ファイル・ライブラリーを示す指定された *File* 内のシンボルの情報を表示します。ファイルにシンボル情報がない場合、**nm** コマンドはその事実を報告しますが、それをエラー条件としては解釈しません。デフォルトでは、**nm** コマンドは数値を 10 進表記で報告します。

**nm** コマンドは、次のシンボル情報を標準出力に書き出します。

- ライブラリーまたはオブジェクト名

**nm** コマンドは、**-A** オプションが指定されている場合のみ、ファイルに関連するライブラリーまたはオブジェクト名のいずれかを報告します。

- シンボル名
- シンボル・タイプ

**nm** コマンドは、次のいずれかの文字でファイルのシンボル・タイプを表します (弱いシンボルはグローバル・シンボルと同じ文字で表されます)。

<b>A</b>	グローバル絶対シンボル
<b>a</b>	ローカル絶対シンボル
<b>B</b>	グローバル bss シンボル
<b>b</b>	ローカル bss シンボル
<b>D</b>	グローバル・データ・シンボル
<b>d</b>	ローカル・データ・シンボル
<b>f</b>	ソース・ファイル名シンボル
<b>L</b>	グローバル・スレッド・ローカル・シンボル (TLS)
<b>l</b>	静的スレッド・ローカル・シンボル (TLS)
<b>T</b>	グローバル・テキスト・シンボル
<b>t</b>	ローカル・テキスト・シンボル
<b>U</b>	未定義シンボル

- 値
- サイズ

**nm** コマンドは、利用可能な場合は、シンボルに関連するサイズを報告します。

### フラグ

**-A** 各行のオブジェクトの絶対パス名またはライブラリー名のいずれかを表示します。



- B 「Berkeley Software Distribution (BSD)」フォーマットで出力を表示します。
- value type name
- C C++ 名のデマングリングを抑制します。デフォルトでは、すべての C++ シンボル名がデマングルされます。  
注: C++ オブジェクト・ファイルからのシンボルは、使用する前に名前がデマングルされます。
- d シンボルの値とサイズを 10 進数として表示します。これはデフォルトです。
- e 静的シンボルと外部 (グローバル) シンボルのみを表示します。
- f 通常は抑制される冗長な .text、.data、および .bss シンボルを含む、全出力を表示します。
- g 外部 (グローバル) シンボルのみを表示します。
- h 出力ヘッダー・データの表示を抑制します。
- I WEAK シンボルのキー文字に \* を付加することにより、WEAK シンボルと GLOBAL シンボルを区別します。-P オプションと一緒に使用した場合、弱いシンボルのシンボル・タイプは次のように表されます。  
  
  - V 弱いデータ・シンボル
  - W 弱いテキスト・シンボル
  - w 弱い未定義シンボル
  - Z 弱い bss シンボル
- o シンボルの値とサイズを 10 進数ではなく 8 進数として表示します。
- p ソートしません。出力データは、記号テーブルの順序で表示されます。
- P 標準ポータブル出力フォーマットで情報を表示します。  
  

library/object name	name	type	value	size
このフォーマットでは、-t、-d、または -o フラグを使って別のフォーマットを指定しない限り、数値が 16 進表記で表示されます。				
-P フラグを付けると、-A フラグが指定されている場合のみ、library/object name フィールドが表示されます。また、-P フラグを付けると、サイズを適用できるシンボルに対してのみ、size フィールドが表示されます。				
- r 逆順にソートします。
- t *Format* 指定されたフォーマットで数値を表示します。Format には、次のいずれかの表記を指定します。  
  
  - d 10 進表記。これは、nm コマンドのデフォルトのフォーマットです。
  - o 8 進表記。
  - x 16 進表記。
- T 本来は桁からあふれるすべての名前を切り捨て、名前の最終文字をアスタリスクにします。デフォルトでは、nm はリストされるシンボル名全体を表示し、その隣に桁幅よりも長い名前を表示するので、名前のあとのすべての桁の位置がずれずれます。
- u 未定義シンボルのみを表示します。
- v アルファベットではなく値で出力をソートします。
- x シンボルの値とサイズを 10 進数ではなく 16 進数として表示します。

- X mode** **nm** が調べるオブジェクト・ファイルのタイプを指定します。 *mode* には、次のいずれかを指定します。
- 32** 32 ビットのオブジェクト・ファイルのみを処理します。
  - 64** 64 ビットのオブジェクト・ファイルのみを処理します。
  - 32\_64** 32 ビットと 64 ビットのオブジェクト・ファイルを処理します。
  - d64** 不連続 64 ビット XCOFF ファイル (マジック番号 == U803XTOCMAGIC) を検査します。
  - any** サポートされるオブジェクト・ファイルをすべて処理します。

デフォルトでは、32 ビットのオブジェクトが処理されます (64 ビットのオブジェクト・ファイルは無視されます)。 *mode* は、**OBJECT\_MODE** 環境変数を使用して設定することもできます。例えば、**OBJECT\_MODE=64** を指定すると、**nm** は 64 ビットのオブジェクトを処理し、32 ビットのオブジェクトを無視します。 **-X** フラグは **OBJECT\_MODE** 変数をオーバーライドします。

**注:** **nm** コマンドは、**--** (ダブル・ハイフン) フラグをサポートしています。このフラグは、ファイル名が誤ってオプションとして解釈される可能性がある場合に、*File* オペランドを区別するために使用します。例えば、ハイフンで始まるファイル名を指定する際は、**--** フラグを使用します。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0** 正常終了。
- >0** エラーが発生しました。

## 例

1. オブジェクト・ファイル **a.out** の静的シンボルと外部シンボルをリストするには、次のように入力します。

```
nm -e a.out
```

2. シンボルのサイズと値を 16 進数として表示し、シンボルを値でソートするには、次のように入力します。

```
nm -xv a.out
```

3. **libc.a** 内の 64 ビットのすべてのオブジェクトのシンボルを表示し、32 ビットのオブジェクトをすべて無視するには、次のように入力します。

```
nm -X64 /usr/lib/libc.a
```

## ファイル

**/usr/ccs/bin/nm** **nm** コマンドを含みます。

## 関連情報

**ar** コマンド、**as** コマンド、**ld** コマンド、**size** コマンド、**strip** コマンド。

**a.out** ファイル、**ar** ファイル。

オペレーティング・システムおよびデバイスのマネージのコマンドのセクション。

---

## nmon コマンド

### 目的

対話モードでローカル・システム統計を表示し、記録モードでシステム統計を記録します。

### 構文

対話モード:

```
nmon [ -h ]
```

```
nmon [ -s < second > ] [ -c < count > ] [ -b ] [ -B ] [ -g < filename > ] [ -k disklist ]
```

記録モード:

```
nmon [ -f | -F | -x | -X | -z ] [ -r < runname > ] [ -t | -T | -Y ] [ -s seconds ] [ -c number ] [ -w number ] [ -l dpl ] [ -d ] [ -g filename ] [ -k disklist ] [ -G ] [ -K ] [ -o outputpath ] [ -D ] [ -E ] [ -J ] [ -V ] [ -P ] [ -M ] [ -N ] [ -W ] [ -S ] [ -^ ] [ -O ] [ -L ] [ -l percent ] [ -A ] [ -m < dir > ] [ -Z priority ]
```

注: 記録モードでは、**-f**、**-F**、**-z**、**-x**、または **-X** の各フラグのうちの 1 つだけを最初の引数として指定します。

### 説明

**nmon** コマンドを使用して、ローカル・システム情報の表示と記録を行います。このコマンドには、対話モードと記録モードがあります。 **-F**、**-f**、**-X**、**-x**、および **-Z** フラグのいずれかを指定した場合も、**nmon** コマンドは記録モード状態です。それ以外の場合は、**nmon** コマンドは対話モード状態となります。

**nmon** コマンドを使用すると、対話モードで以下の表示を行うことができます。

- システム・リソース・ビュー (**r** キーを使用)
- プロセス・ビュー (**t** と **u** キーを使用)
- AIO プロセス・ビュー (**A** キーを使用)
- プロセッサ使用状況の小規模ビュー (**c** キーを使用)
- プロセッサ使用状況の大規模ビュー (**C** キーを使用)
- 共用プロセッサの論理区画ビュー (**p** キーを使用)
- NFS パネル (**N** キーを使用)
- ネットワーク・インターフェース・ビュー (**n** キーを使用)
- WLM ビュー (**W** キーを使用)
- ディスク・ビジー・マップ (**o** キーを使用)
- ディスク・グループ (**g** キーを使用)
- ESS vpath 統計ビュー (**e** キーを使用)
- JFS ビュー (**j** キーを使用)
- カーネル統計 (**k** キーを使用)
- 長期間のプロセッサ平均ビュー (**l** キーを使用)
- ラージ・ページ分析 (**L** キーを使用)

- ページ・スペース (**P** キーを使用)
- ボリューム・グループ統計 (**V** キーを使用)
- ディスク統計 (**D** キーを使用)
- グラフ付きディスク統計 (**d** キーを使用)
- メモリーとページング統計 (**m** キーを使用)
- アダプター入出力統計 (**a** キーを使用)
- 共用イーサネット・アダプター統計 (**O** キーを使用)
- 詳細検査 OK/警告/危険のビュー (**v** キーを使用)
- 詳細ページ統計 (**M** キーを使用)
- ファイバー・チャンネル・アダプター統計 (**^** キーを使用)

記録モードでは、このコマンドは **.nmon** ファイルを生成し、そのファイルを **nmon** アナライザーを使用して表示できます。

**nmon** コマンドが開始するたびに同一セットのキーを使用すると、**NMON** シェル変数にキーをセットできます。例えば、以下のコマンドを実行できます。

```
export NMON=mcd
```

次に、**nmon** コマンドを実行します。

コマンド行から **nmon** コマンドを停止するには、**nmon** プロセス ID を指定した **kill -USR2** を使用します。

**nmon** 記録のバックグラウンド・プロセス ID を印刷するには、**-p** フラグ指定で **nmon** コマンドを実行します。

**nmon** コマンドがリストするプロセスを制限するには (オンライン、およびファイルに対して)、環境変数内のプログラム名を **NMONCMD0** から **NMONCMD63** の間で設定するか、または **cmd:cmd:cmd** パラメーター指定で **-C** フラグを使用するかのいずれかを行うことができます。例えば、以下のコマンドを入力できます。

```
nmon -C ksh:vi:syncd
```

**nmon** がリストするディスクを最大 64 までに制限するには (オンラインのみ)、**diskname,diskname,diskname** パラメーター指定で **-k** フラグを使用します。例えば、以下のコマンドを入力できます。

```
nmon -k hdisk2,hdisk0,hdisk3
```

**nmon** ツールは記録中にはシェルから切断されます。これは、お客様がログアウトした場合でもこのコマンドが実行を継続できるようにするためです。

**nmon** でジャーナル・ファイルシステム (JFS) の記録とモニターを行うと、統計収集時にはこのファイルシステムが使用中になるため、ファイルシステムのアンロードを防止できます。

ワークロード・パーティション (WPAR) 内では、**nmon** コマンドはプロセッサとメモリー統計用のグローバル値を表示します。他の値は、WPAR 固有です。WPAR 内部では以下の統計を取り出すことができず、**nmon** 画面は WPAR 内部では以下の統計をサポートしません。

- ディスク、ディスク入出力グラフ、ディスク・ビジー・マップ、ディスク・グループ
- ディスク・アダプター

- ページ・スペース
- ボリューム・グループ
- ESS/vpath
- ファイバー・チャンネル・アダプター
- VIOS 共用イーサネット・アダプター

## 対話モードでのフラグ

対話モードでは以下のフラグを使用できます。

<b>-s</b> < <i>seconds</i> >	ある画面のリフレッシュから次のリフレッシュまでの時間間隔。デフォルト値は 2 秒です。
<b>-c</b> < <i>count</i> >	画面をリフレッシュする回数。
<b>-g</b> < <i>filename</i> >	<i>filename</i> パラメーターを使用するユーザー定義ディスク・グループを含むファイル。ファイルの中の各行はグループ名で始まります。ハード・ディスクのリストは、グループ名の後ろに続き、スペースで区切られます。このファイルには最大 64 個のディスク・グループを入れることができます。1 つのハード・ディスクは、さまざまなディスク・グループに所属することができます。
<b>-b</b>	モノクロ・モードでビューを表示します。
<b>-B</b>	このビューの中にボックスを組み込みません。デフォルトでは、このコマンドはボックスを表示します。
<b>-h</b>	ヘルプ情報を表示します。
<b>-k</b> < <i>disklist</i> >	ディスク・リスト内のディスクのみをレポートします。

## 記録モードでのフラグ

<b>-A</b>	ビューに非同期入出力セクションを組み込みます。
<b>-c</b>	このコマンドに対してスナップショットを取った回数を指定します。デフォルト値は 10000000 です。
<b>-d</b>	ビューにディスク・サービス時間セクションを組み込みます。
<b>-D</b>	ディスク構成セクションをスキップします。
<b>-E</b>	ESS 構成セクションをスキップします。
<b>-f</b>	出力がスプレッドシート形式であることを指定します。デフォルトでは、このコマンドは 300 秒の間隔にデータを 288 回、各スナップショット間で記録します。出力ファイル名は、 <i>hostname_YYMMDD_HHMM.nmon</i> の形式になっています。
<b>-F</b>	出力がスプレッドシート形式であること、および出力ファイルの名前が <i>filename</i> であることを指定します。 <i>filename</i> パラメーターには出力ファイルの名前を指定します。
<b>-g</b>	<i>filename</i> パラメーターを使用して、ユーザー定義ディスク・グループを含むファイルを指定します。ファイルの中の各行はグループ名で始まります。ディスクのリストは、グループ名の後ろに続き、スペースで区切られます。このファイルには最大 64 個のディスク・グループを入れることができます。1 つのディスクは、さまざまなディスク・グループに所属することができます。
<b>-G</b>	現地時間ではなく、グリニッジ標準時 (GMT) を使用します。これが有用なのは、1 つのマシンにある多くの LPAR から <i>nmon</i> ファイルをプロセッサ・ビュー用に比較する場合で、かつ、これらの LPAR が異なるタイム・ゾーンにある場合です。
<b>-I</b>	このコマンドが TOP プロセス統計を無視するプロセスしきい値のパーセンテージを指定します。デフォルトのパーセンテージはゼロです。このプロセスが指定パーセンテージよりも少ないプロセッサを使用している場合は、コマンドは TOP プロセス統計を保存しません。
<b>-J</b>	JFS セクションをスキップします。
<b>-k</b>	記録対象のディスク・リストを指定します。

<b>-K</b>	RAW カーネル・セクションと LPAR セクションを記録ファイルに組み込みます。 <b>-K</b> フラグは対応するデータ構造の未加工の数をダンプします。このダンプは読み取り可能で、コマンドがそのデータを記録しようとする時に使用できます。
<b>-l</b>	各行にリストするディスク数を指定します。デフォルトでは、150 ディスクが 1 行にリストされます。EMC ディスクの場合、64 の値を指定してください。
<b>-L</b>	ラージ・ページ分析セクションを組み込みます。
<b>-m</b>	コマンドがデータをファイルに保存する前に、ディレクトリーを変更します。
<b>-M</b>	記録ファイルに MEMPAGES セクションを組み込みます。MEMPAGES セクションでは、ページ・サイズごとの詳細メモリー統計が表示されます。
<b>-N</b>	記録ファイルに NFS セクションを組み込みます。NFSv4 統計を収集するには、 <b>-NN</b> を指定します。
<b>-o</b>	記録されたファイルの格納先となるファイル名またはディレクトリー名を指定します。
<b>-O</b>	記録ファイルに共用イーサネット・アダプター (SEA) VIOS セクションを組み込みます。
<b>-P</b>	記録ファイルにページング・スペース・セクションを組み込みます。
<b>-r</b>	スプレッドシート・ファイルに書き込まれる <i>runname</i> フィールドの値を指定します。デフォルトでは、この値はホスト名です。
<b>-s</b>	2 つの連続する記録用スナップショットの間の間隔を秒数で指定します。
<b>-S</b>	サブクラス指定の WLM セクションを記録ファイルに組み込みます。
<b>-t</b>	トップ・プロセスを出力に組み込みます。 <b>-t</b> 、 <b>-T</b> 、または <b>-Y</b> フラグと一緒に指定できません。
<b>-T</b>	トップ・プロセスを出力に組み込み、コマンド行引数を UARG セクション保存します。 <b>-t</b> 、 <b>-T</b> 、または <b>-Y</b> フラグと一緒に指定できません。
<b>-V</b>	ディスク・ボリューム・グループ・セクションを組み込みます。
<b>-w</b>	記録対象のタイム・スタンプ (Tnnnn) のサイズを指定します。このタイム・スタンプは、 <b>.csv</b> ファイルに記録されます。 <i>number</i> パラメーター値の許容範囲は、4 から 16 です。NMON アナライザーの場合は、4 または 8 の値を使用してください。
<b>-W</b>	記録ファイルに WLM セクションを組み込みます。
<b>-x</b>	キャパシティー・プランニングとして 1 日に記録する実際的なスプレッドシートを指定します。デフォルトでは、900 秒ごとに記録され、96 回記録されます。このフラグは、 <b>-ft -s 900 -c 96</b> と同じです。
<b>-X</b>	キャパシティー・プランニングとして 1 時間に記録する実際的なスプレッドシートを指定します。デフォルトでは、30 秒ごとに記録され、120 回記録されます。このフラグは、 <b>-ft -s 30 -c 120</b> と同じです。
<b>-Y</b>	一緒に追加され、記録される同一名のコマンドすべてを使って、記録の中にトップ・プロセスを組み込みます。 <b>-t</b> 、 <b>-T</b> 、または <b>-Y</b> フラグと一緒に指定できません。
<b>-z</b>	キャパシティー・プランニングとして 1 日に記録する実際的なスプレッドシートを指定します。デフォルトでは、900 秒ごとに記録され、96 回記録されます。このフラグは、 <b>-f -s 900 -c 96</b> と同じです。
<b>-Z</b>	実行しようとする <b>nmon</b> コマンドの優先順位を指定します。値 <b>-20</b> は重要であることを意味します。値 <b>20</b> は重要でないことを意味します。 <b>root</b> ユーザーだけが負の値を指定できます。
<b>-^</b>	ファイバー・チャンネル (FC) セクションを組み込みます。

## パラメーター

<i>disklist</i>	ディスク・リストを指定します。
<i>dir</i>	ディレクトリーを指定します。
<i>dpl</i>	各行にリストするディスク数を指定します。
<i>filename</i>	選択するディスク・グループを含むファイルを指定します。
<i>number</i>	リフレッシュ回数を指定します。
<i>count</i>	記録する回数を指定します。
<i>percent</i>	プロセッサ使用のパーセンテージを指定します。
<i>priority</i>	実行対象のプロセスの優先順位を指定します。
<i>runname</i>	実行対象のスプレッドシート・ファイルに <i>runname</i> フィールドの値を指定します。



seconds  
outputpath

スナップショットをリフレッシュする間隔 (秒単位) を指定します。  
出力ファイルのパスを指定します。

## サブコマンド

スペース	画面を即時にリフレッシュします。
.	ビジー・ディスクとプロセスだけを表示します。
~	topas 画面に切り替えます。
^	ファイバー・チャネル・アダプター統計を表示します。
+	画面リフレッシュ時間を 2 倍にします。
-	画面リフレッシュ時間を 2 分の 1 にします。
0	統計のピーク値 (画面に表示) をゼロにリセットします。ピーク値を表示するパネルに対してのみ適用されます。
a	アダプターの入出力統計を表示します。
A	非同期入出力 (aioserver) プロセスを要約します。
b	モノクロ・モードでビューを表示します。
c	棒グラフを使ってプロセッサ統計を表示します。
C	プロセッサ統計を表示します。この統計が有用なのは、プロセッサ数の範囲が 15 から 128 の場合の比較に対してです。
d	ディスクの入出力情報を表示します。特定のディスクのみを表示するには、 <b>-k</b> フラグを指定します。
D	ディスクの入出力統計を表示します。ディスクの統計をもっと表示するには、 <b>D</b> キーを複数回押します。
e	ESS 仮想バーチャルの論理ディスクの入出力統計を表示します。
g	ディスク・グループの入出力統計を表示します。このキーと一緒に <b>-g</b> フラグを指定する必要があります。
h	オンライン・ヘルプ情報を表示します。
j	JFS 統計を表示します。
k	カーネルの内部統計を表示します。
l	長形式のプロセッサ統計を表示します。75 を超えるスナップショットを棒グラフを使って表示します。
m	メモリとページング統計を表示します。
M	複数のページ・サイズ統計をページ数単位で表示します。 <b>M</b> キーを 2 回押すと、この統計がメガバイト単位で表示されます。
n	ネットワーク統計を表示します。
N	NFS ネットワーク・ファイルシステム統計を表示します。 <b>N</b> キーを 2 回押すと、NFSv4 統計が表示されます。
o	ディスク入出力マップが表示されます。
O	共用イーサネット・アダプター VIOS のみを表示します。
p	区画の統計を表示します。
P	ページング・スペースの統計を表示します。
q	終了。 <b>x</b> 、または <b>Ctrl+C</b> キー・シーケンスを使用しても構いません。
r	リソース・タイプ、マシン名、キャッシュ詳細、AIX バージョン、および LPAR 情報を表示します。
S	サブクラス指定の WLM を表示します。
t	トップ・プロセスの統計を表示します。このサブコマンドと一緒に、以下のキーを押すことができます。

- 1: 基本詳細の表示。
- 2: 累積プロセス情報の表示。
- 3: ビューをプロセッサ別にソート。
- 4: ビューをサイズ別にソート。
- 5: ビューを入出力情報別にソート。



<b>u</b>	コマンド引数を指定してトップ・プロセスを表示します。新規のプロセス用に引数をリフレッシュするには、 <b>u</b> キーを 2 回押します。
<b>U</b>	コマンド引数を指定してトップ・プロセスを表示し、ワークロード・クラスまたはワークロード・パーティション情報を表示します。
<b>v</b>	定義済みシステム・リソースの現行の状況を強調表示し、危険、警告、または正常のいずれかとしてそのシステム・リソースを分類します。
<b>V</b>	ディスク・ボリューム・グループの統計を表示します。
<b>w</b>	トップ・プロセスにより使用された時点での待ち状態のプロセスを表示します。
<b>W</b>	ワークロード・マネージャー (WLM) の統計を表示します。

## 出力詳細

このセクションでは、nmon 画面上に表示されるメトリックの説明を記述します。

### システム・リソース・ビュー

このビューでは、システム・リソースの一般情報が表示されます。このビューを表示するには、**r** キーを押します。このビューには、以下のリソース情報が含まれます。

- システム内のプロセッサ数。
- システム内でアクティブ状態のオンライン・プロセッサ数。
- プロセッサの周波数。
- AIX のバージョンとそのテクニカル・レベル。
- 実行中のカーネルのタイプ。
- 論理区画。
- ハードウェア・モデル。
- システムのプロセッサ・アーキテクチャー。
- プラットフォーム・バスのタイプ。
- プロセッサのキャッシュ情報。
- アクティブ・イベント数。
- 以前のシリアル番号。この番号は、動的構成イベント以前の区画のマシン ID です。
- 現行のシリアル番号。この番号は、現行のマシン ID、または動的構成イベント以降の区画のマシン ID です。
- 最後の動的再構成イベントが発生した現地時間。この情報は、"When" キーワードと一緒にラベル付けされます。

### AIO プロセス・ビュー

AIO プロセス・ビューでは、非同期入出力 (AIO) プロセス情報が表示されます。このビューを表示するには、**A** キーを押します。以下のカラムが画面上に表示されます。

<b>Total AIO Processes</b>	AIO プロセスの総数。
<b>Actually in use</b>	プロセッサの 0.1% を超える量を消費している AIO プロセス数。
<b>CPU Used</b>	カーネル・プロセスのすべてが使用したプロセッサのパーセンテージ。
<b>All time peak</b>	システムの始動以降に実行しているカーネル・プロセスの最大数。
<b>Recent peak</b>	プロセッサの 0.1% を超える量を消費しているカーネル・プロセスの最近の最大数。
<b>Peak</b>	カーネル・プロセスのすべてが使用したプロセッサの最大パーセンテージ。

## プロセス・ビュー

プロセス・ビューには、システムでのプロセス詳細が表示されます。このビューを表示するには、**t** キーまたは **v** キーを押します。このビューでは、以下のカラムが画面上に表示されます。

<b>pid</b>	プロセスの ID。
<b>ppid</b>	親プロセスの ID。
<b>User</b>	プロセスのユーザー ID。
<b>Proc Group</b>	プロセス・グループの ID。
<b>Nice</b>	プロセスの初期優先順位。この値は、 <b>nice</b> コマンドにより設定されます。
<b>Priority</b>	プロセスの基本スケジューリング優先順位。
<b>Status</b>	プログラムの状況。
<b>Proc_Flag</b>	プロセスのフラグ。
<b>Thrds</b>	スレッド数。
<b>Files</b>	使用中の最大ファイル索引。
<b>Foreground</b>	フォアグラウンド・プロセスまたはバックグラウンド・プロセス。
<b>Command</b>	コマンド名。
<b>Time Start</b>	コマンドが開始した時刻。
<b>CPU-Total</b>	プロセスの開始以降にそのプロセスが要した合計時間。
<b>Child Total</b>	子プロセスの開始以降にそのプロセスが要した合計時間。
<b>Delta-Total</b>	その間隔の中でそのプロセスが消費した合計時間。
<b>%CPU Used</b>	最後の間隔で消費したプロセッサのパーセンテージ。
<b>Size KB</b>	ページのサイズ (キロバイト単位)。
<b>Res Size</b>	実メモリー・データ (常駐セット) の合計、およびプロセスの実メモリー (常駐セット) 文字サイズの合計。
<b>Res Set</b>	実メモリー・データ (常駐セット) の合計、およびプロセスの実メモリー (常駐セット) 文字サイズの合計。
<b>Res Text</b>	プロセスの実メモリー文字サイズ。
<b>Res Data</b>	プロセスの実メモリー・データ・サイズ。
<b>Char I/O</b>	最後の間隔での入出力文字数/秒。
<b>RAM Use</b>	使用された RAM のパーセンテージ。
<b>Paging I/O</b>	最後の間隔での入出力ページ不在数/秒。
<b>Paging Other</b>	最後の間隔での非入出力ページ不在数/秒。
<b>Paging Repages</b>	最後の間隔での再ページ不在数/秒。
<b>Class</b>	プロセスの ワークロード・マネージャー クラス名。

## プロセッサ使用の小規模ビュー

プロセッサ使用の小規模ビューでは、次の概要 (ユーザー、システム、論理プロセッサのアイドル時間と待ち時間、対応ライセンス、および仮想プロセッサ使用量) を表示します。プロセッサ使用の小規模ビューを生成するには、**c** キーを使用します。

## プロセッサ使用の大規模ビュー

プロセッサ使用の大規模ビューでは、論理プロセッサの使用状況をグラフで表示します。このビューを表示するには、**C** キーを押します。

以下のラベルを使用して異なるモードで消費された時間を明確化します。

- **s**: システム・モードで消費された時間のパーセンテージを指定します。
- **u**: ユーザー・モードで消費された時間のパーセンテージを指定します。

## 共用プロセッサ論理区画ビュー

共用プロセッサ論理区画ビューには、区画に関する以下の情報を示すフラグが含まれます。

- 区画が LPAR かどうか

- 区画が LPAR になることができるかどうか
- 区画が共用か専用のどちらか
- SMT がオンかオフかどうか
- 共用区画がキャップかアンキャップかどうか

このビューを表示するには、**p** キーを押します。

#### プロセッサ:

プロセッサ状況に関する以下のメトリックがこのビューに表示されます。

<b>Max Phys in Sys</b>	システム内の物理プロセッサの最大数
<b>Phys CPU in system</b>	システム内の物理プロセッサ数
<b>Virtual Online</b>	オンラインの仮想プロセッサ数
<b>Logical online</b>	オンラインの論理プロセッサ数
<b>Physical pool</b>	この区画に割り当てられた、共用プール ID にある共用物理プロセッサ数
<b>SMT threads/CPU</b>	SMT スレッド数/プロセッサ

#### キャパシティー:

プロセッサに関する以下のキャパシティー情報が表示されます。

<b>Cap. Processor Min</b>	この LPAR 用に定義された処理単位の最小数
<b>Cap. Processor Max</b>	この LPAR 用に定義された処理単位の最大数
<b>Cap. Increment</b>	ライセンス済みキャパシティーへの変更を行うことができる細分度
<b>Cap. Unallocated</b>	LPAR グループ内の共用 LPAR から割り振り解除されたプロセッサ・ユニット数の合計
<b>Cap. Entitled</b>	ライセンス済みキャパシティー
<b>MinReqVirtualCPU</b>	この LPAR に対して最低限必要な仮想プロセッサ数

#### ID メモリー:

ID メモリーの以下のメトリックが表示されます。

<b>LPAR ID Group:Pool</b>	LPAR グループの ID とそのプール ID
<b>Memory (MB/GB) Min:Max</b>	この LPAR に対して定義された最小と最大のメモリー (メガバイト単位またはギガバイト単位)
<b>Memory(MB/GB) Online</b>	オンラインの実メモリー (メガバイト単位またはギガバイト単位)
<b>Memory Region LMB</b>	1 論理メモリー・ブロック (LMB) のサイズ (バイト単位)

#### 時間(秒単位):

<b>Time Dispatch Wheel</b>	各仮想プロセッサがそのライセンスを受け付ける間隔
<b>MaxDispatch Latency</b>	物理プロセッサ上で、LPAR のディスパッチから次のディスパッチの間の最大待ち時間 (秒単位)
<b>Time Pool Idle</b>	共用プロセッサ・プールがアイドルとなる時間 (秒単位)
<b>Time Total Dispatch</b>	LPAR がディスパッチする合計時間 (秒単位)

#### プロセッサ数の最小値と最大値

プロセッサ数に関する以下の最小値と最大値が表示されます。

<b>Virtual CPU ( Min - Max )</b>	LPAR 定義にある仮想プロセッサの最小数と最大数
<b>Logical CPU ( Min - Max )</b>	論理プロセッサの最小数と最大数

## 重み付け

プロセッサの重み付けに関する以下の情報が表示されます。

<b>Weight Variable</b>	プロセッサ・キャパシティーに関する可変の重み付け
<b>Weight Unallocated</b>	この区画に使用可能な未割り振りの可変の重み付け

## NFS パネル

NFS パネルでは、ネットワーク・ファイルシステム (NFS) に関する統計情報が表示されます。このビューを表示するには、**N** キーを押します。このビューには以下のメトリックが含まれています。

<b>Root</b>	NFS V2 サーバーとクライアントのルート要求
<b>Wrccache</b>	NFS サーバーとクライアントの書き込みキャッシュ要求
<b>Null</b>	NFS サーバーとクライアントの書き込みキャッシュ要求
<b>Getattr</b>	NFS サーバーとクライアントの属性獲得要求
<b>Setattr</b>	NFS サーバーとクライアントの属性設定要求
<b>Lookup</b>	NFS サーバーとクライアントのファイル名検索要求
<b>Readlink</b>	NFS サーバーとクライアントのリンク読み取り要求
<b>Read</b>	NFS サーバーとクライアントの読み取り要求
<b>Write</b>	NFS サーバーとクライアントの書き込み要求
<b>Create</b>	NFS サーバーとクライアントのファイル作成要求
<b>Mkdir</b>	NFS サーバーとクライアントのディレクトリ作成要求
<b>Symlink</b>	NFS サーバーとクライアントのシンボリック・リンク作成要求
<b>Remove</b>	NFS サーバーとクライアントのファイル除去要求
<b>Rmdir</b>	NFS サーバーとクライアントのディレクトリ除去要求
<b>Rename</b>	NFS サーバーとクライアントのファイル名前変更要求
<b>Link</b>	NFS サーバーとクライアントのリンク作成要求
<b>Readdir</b>	NFS サーバーとクライアントのディレクトリ読み取り要求
<b>Fsstat</b>	NFS サーバーとクライアントのファイル状況要求
<b>Access</b>	NFS V3 サーバーとクライアントのアクセス要求
<b>Mknod</b>	NFS V3 サーバーとクライアントの <b>mknod</b> 作成要求
<b>readdir+</b>	NFS V3 サーバーとクライアントのディレクトリ読み取りプラス要求
<b>Fsinfo</b>	NFS V3 サーバーとクライアントのファイル情報要求
<b>Pathconf</b>	NFS V3 サーバーとクライアントのパス構成要求
<b>Commit</b>	NFS サーバーとクライアントのコミット要求
<b>Bad calls</b>	NFS サーバーとクライアントの失敗した呼び出し
<b>Calls</b>	NFS サーバーとクライアントの要求

以下の NFS V4 クライアント/サーバー統計が表示されるのは、**N** キーを 2 回押した場合です。

<b>Access</b>	NFS V4 サーバーとクライアントのアクセス要求
<b>acl_read</b>	NFS V4 クライアントの読み取りアクセス制御リスト (ACL)
<b>acl_stat_l</b>	NFS V4 クライアントの長 ACL 情報取得
<b>acl_write</b>	NFS V4 クライアントの書き込みアクセス制御リスト (ACL)
<b>Clntconfirm</b>	NFS V4 クライアントの確認操作
<b>Close</b>	NFS V4 クライアントのファイル・クローズ
<b>Commit</b>	NFS V4 サーバーとクライアントをコミット
<b>Compound</b>	NFS V4 サーバーの複合呼び出し

<b>Create</b>	NFS V4 サーバーとクライアントの標準外オブジェクト作成
<b>DeleGPurge</b>	NFS V4 サーバーが代行待ちリカバリーを消去
<b>DeleGReturn</b>	NFS V4 サーバーとクライアントの代行戻し
<b>Finfo</b>	NFS V4 クライアントのファイル情報入手
<b>getattr</b>	NFS V4 サーバーとクライアントの属性取得
<b>getfh</b>	NFS V4 サーバーのファイル・ハンドル取得
<b>Link</b>	NFS V4 サーバーとクライアントのリンク操作
<b>Lock</b>	NFS V4 サーバーとクライアントのロック操作
<b>lockt/test</b>	NFS V4 サーバーが指定のロックをテストまたは NFS V4 クライアントのロック・テスト
<b>locku/unlock</b>	NFS V4 サーバーまたは NFS V4 クライアントのアンロック操作
<b>lookup</b>	NFS V4 サーバーとクライアントのファイル名検索
<b>lookupp</b>	NFS V4 サーバーの親ディレクトリー検索
<b>mkdir</b>	NFS V4 クライアントのディレクトリー作成
<b>mknod</b>	NFS V4 クライアントのスペシャル・ファイル作成
<b>Null</b>	NFS V4 サーバーのヌル呼び出しまたは NFS V4 クライアントのヌル呼び出し
<b>nverify</b>	NFS V4 サーバーの属性内の相違検査
<b>openattr</b>	指定された属性ディレクトリーを NFS V4 サーバーがオープン
<b>openconfirm</b>	NFS V4 サーバーとクライアントが、使用のためのオープン確認
<b>opendowngrade</b>	NFS V4 サーバーとクライアントが指定ファイルのアクセス権限を格下げ
<b>Open</b>	NFS V4 サーバーとクライアントのオープン操作
<b>operations</b>	NFS V4 サーバーとクライアントの操作
<b>pcl_read</b>	NFS V4 クライアントが、プリンター・コントロール言語 (PCL) ファイルから数値データを抽出
<b>pcl_readstat_l</b>	NFS V4 クライアントの <b>pcl_stat</b> 詳細操作
<b>pcl_stat</b>	NFS V4 クライアントの <b>pcl_stat</b> 操作
<b>pcl_write</b>	NFS V4 クライアントの <b>pcl_write</b> 操作
<b>putfh</b>	NFS V4 サーバーの現行ファイル・ハンドル設定
<b>putpubfh</b>	NFS V4 サーバーの一般ファイル・ハンドル設定
<b>putrootfh</b>	NFS V4 サーバーのルート・ファイル・ハンドル設定
<b>readdir</b>	NFS V4 サーバーとクライアントのディレクトリー読み取り
<b>readlink</b>	NFS V4 サーバーとクライアントのシンボリック・リンク読み取り
<b>Read</b>	NFS V4 サーバーとクライアントがファイルからデータを読み取り
<b>release</b>	NFS V4 サーバーとクライアントの <b>release_lock</b> 操作
<b>remove</b>	NFS V4 サーバーとクライアントがファイルシステム・オブジェクトを除去
<b>rename</b>	NFS V4 サーバーとクライアントのオブジェクト名変更
<b>renew</b>	NFS V4 サーバーとクライアントのリース更新
<b>replicate</b>	NFS V4 クライアントの複製操作
<b>restorefh</b>	NFS V4 サーバーのファイル・ハンドル復元
<b>rmdir</b>	NFS V4 クライアントのディレクトリー除去
<b>savefh</b>	NFS V4 サーバーのファイル・ハンドル保存
<b>secinfo</b>	NFS V4 サーバーとクライアントがセキュリティ情報を入手
<b>setattr</b>	NFS V4 サーバーとクライアントがオブジェクト属性を設定
<b>setclient</b>	NFS V4 サーバーとクライアントの <b>setclient</b> 操作
<b>statfs</b>	NFS V4 クライアントのファイル統計情報要求
<b>symlink</b>	NFS V4 クライアントのシンボリック・リンク操作
<b>verify</b>	NFS V4 クライアントが同じ属性を検査
<b>write</b>	NFS V4 サーバーとクライアントのファイル書き込み

## ネットワーク・インターフェース・ビュー

ネットワーク・インターフェース・ビューには、ネットワーク・エラー統計が表示されます。この情報を表示するには、**n** キーを押します。

ネットワーク・エラーが発生せずに画面が 3 回更新されると、ネットワーク・インターフェース・ビューにはネットワーク・エラー統計が含まれません。

以下のメトリックがこのビューに表示されます。

<b>I/F Name</b>	インターフェース名
<b>Recv-KB/s</b>	この間隔の中で受信されたデータ (キロバイト/秒)
<b>Trans-KB/s</b>	この間隔の中で送信されたデータ (キロバイト/秒)
<b>Packin</b>	この間隔の中で受信されたパケット数
<b>Packout</b>	この間隔の中で送信されたパケット数
<b>Insize</b>	この間隔の中で受信された平均パケット・サイズ
<b>Outsize</b>	最後の間隔の中で送信された平均パケット・サイズ
<b>Peak-&gt;Recv</b>	受信データのピーク値 (キロバイト/秒)
<b>Peak-&gt;Trans</b>	送信データのピーク値 (キロバイト/秒)
<b>Total Recv</b>	合計受信データ (メガバイト/秒)
<b>Total Sent</b>	合計送信データ (メガバイト/秒)
<b>MTU</b>	トランスポート・ユニットの最大サイズ (バイト単位)
<b>Ierror</b>	入力エラー数
<b>Oerror</b>	出力エラー数
<b>Collision</b>	衝突回数
<b>Mbits/s</b>	アダプター・ビット・レート (メガビット/秒)
<b>Description</b>	このインターフェースの説明

## WLM ビュー

WLM ビューにはワークロード・マネージメントに関する情報が表示されます。このビューを表示するには、**W** キーを使用します。サブクラス・セクションをオンにするには、**WLM** ビューで **S** キーを押します。サブクラス・セクションをオフにするには、再度、**S** キーを押します。

以下のメトリックがこのビューに表示されます。

<b>CPU</b>	そのクラスのプロセッサ使用のパーセンテージ
<b>MEM</b>	そのクラスの物理メモリー使用のパーセンテージ
<b>BIO</b>	そのクラスに対応したディスク入出力の処理能力使用のパーセンテージ
<b>Process (Procs)</b>	そのクラス内のプロセス数
<b>Tier (T)</b>	層番号。この値の範囲は 0 から 9 です。
<b>Inheritance (I)</b>	継承属性の値。0 の値は、no を表し、1 の値は yes を表します。
<b>Location</b>	ロケーションの値。1 の値の意味は、共用クラスへセグメントを転送しないことを表します。そうでない場合、0 の値が表示されます。

## ディスク・ビジー・マップ

ディスク・ビジー・マップにはディスク使用統計が表示されます。このマップを表示するには、**o** キーを押します。1 画面当たり、最大 100 ディスクが表示されます。hdisk0 から hdisk100 までの範囲の名前が付いたディスクのみ表示されます。下表には名前前の範囲に対応したシンボルを記載してあります。

シンボル	名前
—	5 未満
.	10 未満
-	20 未満
+	30 未満
o	40 未満
0	50 未満
O	60 未満
8	70 未満
X	80 未満
#	90 未満



シンボル	名前
@	100 以下

## ディスク・グループ

複数のディスクをモニターするには、そのディスクをグループに入れます。このビューを表示するには、**g** キーを押します。

以下の例に記載した行を含むグループ構成ファイルを作成する必要があります。

```
<Group_name1> <disk_name1> <disk_name2> ....  
<Group_name2> <disk_nameA> <disk_nameB> ...
```

例えば、<Group\_name1> はグループの最初のディスクの名前であり、<disk\_name1> と <disk\_name2> はグループ内の 1 番目と 2 番目のディスクです。

ディスク・グループ入出力を表示するには、**-g** フラグとグループ・ファイルを指定して **nmon** コマンドを実行してから、**-g** キーを押します。以下のメトリックがこのビューに表示されます。

<b>Name</b>	ディスク・グループ名。最大 64 グループを指定できます。1 つのディスクが複数のグループにあっても構いません。
<b>Disks</b>	グループにあるディスク数。
<b>Read/Write-KB/s</b>	この間隔中での読み取りと書き込みのデータ転送速度 (キロバイト/秒)
<b>TotalMB/s</b>	この間隔中での読み取りと書き込みの合計データ (メガバイト/秒)
<b>Xfers/s</b>	この間隔中での読み取りと書き込みのデータ転送回数
<b>BlockSizeKB</b>	1 転送操作当たりの読み取りまたは書き込みのブロック・サイズ (キロバイト単位)

## ESS Vpath 統計ビュー

このビューでは ESS Vpath 統計を表示します。このビューを表示するには、**e** キーを押します。このビューには以下のメトリックが含まれています。

<b>Name</b>	仮想パスの名前。
<b>Size</b>	ESS パスのサイズ。
<b>AvgBusy</b>	このディスクの平均のビジー使用。
<b>Write-KB/s</b>	この間隔中での書き込みデータ転送率 (キロバイト/秒)
<b>Read-KB/s</b>	この間隔中での読み取りデータ転送率 (キロバイト/秒)
<b>Xfers/s</b>	読み取りおよび書き込みの転送回数/秒
<b>Total vpaths</b>	仮想パス数。

## JFS ビュー

このビューには、ジャーナル・ファイルシステム (JFS) 統計が表示されます。このビューを表示するには、**j** キーを押します。以下の統計情報がこのビューに記録されます。

<b>FileSystem</b>	ファイルシステム名。
<b>Size (MB)</b>	ファイルシステムのサイズ (メガバイト)。
<b>Free (MB)</b>	ファイルシステム内の使用可能なフリー・スペース (メガバイト)。
<b>%Used</b>	使用済みのファイルシステムのパーセント。
<b>%Inodes</b>	i-node で使用されているファイルシステムのパーセント。
<b>Mount point</b>	ローカル・マウント・ポイント。



## カーネル統計

このビューにはカーネルの統計情報が表示されます。このビューを表示するには、**k** キーを押します。以下の統計情報がこのビューに表示されます。

<b>runqueue</b>	作動可能状態ではあるが、プロセッサが使用可能になるのを待っているスレッドの平均数。
<b>pswitch</b>	この間隔の中でのプロセッサ・スイッチ回数/秒。
<b>fork</b>	この間隔の中での <b>fork</b> の回数/秒。
<b>exec</b>	この間隔の中での <b>exec</b> の回数/秒。
<b>msg</b>	この間隔の中での送信と受信のプロセス間通信 (IPC) メッセージ数/秒
<b>sem</b>	この間隔の中でのセマフォ・オペレーション・システム呼び出し回数/秒。
<b>hw intrp</b>	この間隔の中でのデバイス割り込みの回数/秒。
<b>sw intrp</b>	この間隔の中でオフ・レベル・ハンドラーが呼び出された回数/秒。
<b>Swapin</b>	この間隔の中でスワップ・キュー内のプロセス数/秒。
<b>Syscall</b>	この間隔の中でのシステム呼び出し回数/秒。
<b>read</b>	この間隔の中での読み取り呼び出し回数/秒。
<b>write</b>	この間隔の中での書き込み呼び出し回数/秒。
<b>readch</b>	この間隔の中での読み取りシステム呼び出しにより転送された文字数。
<b>Writch</b>	この間隔の中での書き込みシステム呼び出しにより転送された文字数。
<b>R + W (MB/s)</b>	この間隔の中での読み取りと書き込みの文字数 (メガバイト/秒)
<b>Uptime</b>	このシステムが稼働状態にあった時間。
<b>iget</b>	この間隔の中での <b>inode</b> ルックアップ回数/秒。
<b>dirblk</b>	この間隔の中で、ファイルのエントリーを探すためにディレクトリー・サーチ・ルーチンによる 512 バイト・ブロック読み取り回数/秒。
<b>namei</b>	この間隔の中でのパス名からの <b>vnode</b> 検索回数/秒。
<b>ksched</b>	この間隔の中でのカーネル・プロセス作成回数/秒。
<b>koverf</b>	この間隔の中でのカーネル・プロセスの作成試行回数/秒であり、この試行回数は、ユーザーが上限まで <b>fork</b> したか、またはプロセスの構成限界に到達した場合の回数です。
<b>kexit</b>	この間隔の中でゾンビになったカーネル・プロセス数/秒。

## 長期間のプロセッサ平均ビュー

このビューでは、瞬間的なシステムに関する情報を表示します。このビューを表示するには、**I** キーを押します。以下のラベルを使用して、異なるモードで消費された時間を明確化します。

- **s**: システム・モードで消費された時間のパーセンテージを指定します。
- **u**: ユーザー・モードで消費された時間のパーセンテージを指定します。
- **w**: 待機モードで消費された時間のパーセンテージを指定します。

以下のメトリックがこのビューに表示されます。

<b>EntitledCPU</b>	区画のライセンス済みキャパシティー。
<b>UsedCPU</b>	その区画で消費される物理プロセッサ数。

## ページ・ページ分析

このビューでは、ラージ・ページの分析が表示されます。このビューを表示するには、**L** キーを押します。以下の情報が表示されます。

<b>Count</b>	ラージ・ページ数およびその合計サイズ。
<b>Free</b>	空きのラージ・ページ数およびそのサイズのパーセンテージ。
<b>In Use</b>	使用中のラージ・ページ数およびそのサイズのパーセンテージ。
<b>Size</b>	ラージ・ページのサイズ。
<b>High water mark</b>	ラージ・ページの高水準点。

## ページング・スペース

このビューではページング・スペース統計を表示します。このビューを表示するには、**p** キーを押します。このビューには以下のメトリックが表示されます。

<b>PagingSpace</b>	ページ・スペース数。
<b>Volume-Group</b>	ボリューム・グループ数。
<b>Type</b>	論理ボリュームのタイプ。このタイプとしては NFS または LV があります。
<b>LPs</b>	論理区画サイズ。
<b>MB</b>	サイズ (メガバイト単位)。
<b>Used</b>	ボリューム・グループの使用パーセンテージ。
<b>IOpending</b>	ページング・スペース内の保留入出力数。
<b>Active/Inactive</b>	アクティブまたは非アクティブのページング・スペース。
<b>Auto/NotAuto</b>	ページング・スペースが自動的にロードされるかどうかを示します。

## ボリューム・グループ統計

このビューには、ボリューム・グループの統計が表示されます。このビューを表示するには、**V** キーを押します。このビューには以下の情報が表示されます。

<b>Name</b>	ボリューム・グループ名。
<b>Disks</b>	グループにあるディスク数。
<b>AvgBusy</b>	ボリューム・グループにあるディスクの平均ビジー。
<b>Read/Write-KB/s</b>	この間隔の中での読み取りと書き込みのデータ転送速度 (キロバイト/秒)
<b>TotalMB/s</b>	この間隔の中での読み取りと書き込みの合計データ (メガバイト/秒)
<b>Xfers/s</b>	この間隔の中での読み取りと書き込みの転送回数
<b>BlockSizeKB</b>	この間隔の中での読み取りまたは書き込みブロック・サイズ/転送 (キロバイト/秒)

## ディスク統計

このビューにはディスクの統計が表示されます。このビューを表示するには、**D** キーを押します。各種メトリックを表示するには、**D** キーを以下の回数だけ押します。

- 1 回: ディスク番号を表示します。
- 2 回: ディスク説明を表示します。
- 3 回: サービス時間を表示します。
- 4 回: **d** キーを押した時に表示されるグラフに類似したグラフと共に、ディスク統計が表示されます。

ディスク番号 (**D** キーを 1 回押す)

以下のメトリックがこのビューに表示されます。

<b>Name</b>	ディスクの名前。
<b>Busy</b>	そのディスクの平均ビジー。
<b>Read-KB/s</b>	この間隔の中での読み取りのデータ転送速度 (キロバイト/秒)
<b>Write-KB/s</b>	この間隔の中での書き込みのデータ転送速度 (キロバイト/秒)
<b>Transfers/sec</b>	この間隔の中での読み取りと書き込みの転送回数
<b>SizeKB</b>	この間隔の中での読み取りまたは書き込みブロック・サイズ/転送 (キロバイト/秒)
<b>Peak</b>	平均ビジーのピーク・パーセンテージ。
<b>Peak KB/s</b>	ピークの読み取りと書き込みデータ (キロバイト/秒)
<b>qDepth</b>	ディスクへの送信要求数で、完了していない送信要求数。
<b>Totals Size (GB)</b>	ディスクの合計サイズ (ギガバイト)。
<b>Totals Free (GB)</b>	ディスク上に残っている合計フリー・スペース (ギガバイト)。
<b>Totals Read (MB/s)</b>	すべてのディスクから読み取られたデータの合計データ転送速度 (メガバイト/秒)
<b>Totals Write (MB/s)</b>	すべてのディスクへ書き込まれたデータの合計データ転送速度 (メガバイト/秒)

## Disk 記述 (D キーを 2 回押す)

以下のメトリックがこのビューに表示されます。

<b>Name</b>	ディスク名。
<b>Size (GB)</b>	ディスク・サイズ (ギガバイト)。
<b>Free (GB)</b>	ディスクに残っているフリー・スペース (ギガバイト)。
<b>Disk Paths</b>	ディスクに定義されたパス数。
<b>Disk Adapter</b>	ディスク・アダプター名。
<b>Volume Group</b>	ディスクが所属するボリューム・グループ。
<b>Disk Description</b>	ディスクの記述。
<b>Totals Size (GB)</b>	ディスクの合計サイズ (ギガバイト)。
<b>Totals Free (GB)</b>	ディスク上に残っている合計フリー・スペース (ギガバイト)。
<b>Totals Read (MB/s)</b>	すべてのディスクから読み取られたデータの合計データ転送速度 (メガバイト/秒)
<b>Totals Write (MB/s)</b>	すべてのディスクへ書き込まれたデータの合計データ転送速度 (メガバイト/秒)

## サービス時間 (D キーを 3 回押す)

このビューには以下のメトリックが表示されます。

<b>Disk</b>	ディスク名
<b>Service (in msecs)</b>	平均サービス時間/要求 (ミリ秒)。
<b>Wait (in msecs)</b>	平均待ち時間/要求 (ミリ秒)。
<b>ServQ size</b>	サービス・キュー内の平均要求数。
<b>WaitQ size</b>	完了待ちになっている平均要求数。
<b>ServQ Full</b>	入ってくる要求をそのディスクが受け入れようとしない回数。
<b>Totals Size (GB)</b>	ディスクの合計サイズ (ギガバイト)。
<b>Totals Free (GB)</b>	ディスク上に残っている合計フリー・スペース (ギガバイト)。
<b>Totals Read (MB/s)</b>	すべてのディスクから読み取られたデータの合計データ転送速度 (メガバイト/秒)
<b>Totals Write (MB/s)</b>	すべてのディスクへ書き込まれたデータの合計データ転送速度 (メガバイト/秒)

## グラフ表示のディスク統計

このビューにはグラフ付きのディスク統計が表示されます。このビューを表示するには、**d** キーを押します。以下のメトリックがこのビューに表示されます。

<b>Name</b>	ディスク名
<b>Busy</b>	このディスクの平均ビジー・パーセンテージ。
<b>Read-KB/s</b>	読み取りのデータ転送速度 (キロバイト/秒)
<b>Write-KB/s</b>	書き込みのデータ転送速度 (キロバイト/秒)

## メモリーとページング統計

このビューにはメモリーとページング統計が表示されます。このビューを表示するには、**m** キーを押します。このビューには以下のメトリックが含まれています。

<b>%Used</b>	物理メモリーとページング・スペース内の使用済みスペースのパーセンテージ。
<b>%Free</b>	物理メモリーとページング・スペース内のフリー・スペースのパーセンテージ。
<b>MB Used</b>	使用状態の物理メモリーとページング・スペース (メガバイト)。
<b>MB Free</b>	空き状態の物理メモリーとページング・スペース (メガバイト)。
<b>Pages/sec to Paging Space</b>	ページング・スペースとの間で転送された入出力ページ数/秒。

<b>Pages/sec to File System</b>	ファイルシステムとの間で転送された入出力ページ数/秒。
<b>Page Scans</b>	クロックによるページ・スキャン回数。
<b>Page Faults</b>	ページ・フォールト数。
<b>Page Cycles</b>	ページ置換サイクル数。
<b>Page Steals</b>	ページ・スチール数。
<b>Numperm</b>	ファイルごとに使用されているフレーム数 (4 KB ページ単位)。
<b>Process</b>	プロセス・セグメントで使用された実メモリーのパーセンテージ。
<b>System</b>	システム・セグメントで使用された実メモリーのパーセンテージ。
<b>Free</b>	空き状態の実メモリーのパーセンテージ。
<b>Total</b>	使用された合計の実メモリーのパーセンテージ。
<b>Min/Maxperm</b>	ページ・スチールに対する <b>minperm</b> と <b>maxperm</b> 値。
<b>Min/Maxfree</b>	<b>minfree</b> と <b>maxfree</b> のページ・フリー・リスト。
<b>Min/Maxpgahead</b>	ページ・アヘッドのページの最小数と最大数。
<b>Total Virtual</b>	合計仮想メモリー。
<b>Accessed Virtual</b>	活動状態の仮想メモリー。
<b>Numclient</b>	クライアント・フレーム数。
<b>Maxclient</b>	クライアント・フレームの最大数。
<b>User</b>	非システム・セグメントが使用する実メモリー。
<b>Pinned</b>	ピンされた実メモリー。

## アダプター入出力統計ビュー

このビューではアダプター入出力統計を表示します。このビューを表示するには、**a** キーを押します。以下のメトリックがこのビューに表示されます。

<b>Adapter</b>	アダプター名。
<b>Busy%</b>	アダプターの帯域幅使用。
<b>Read-KB/s</b>	読み取りのデータ転送速度 (キロバイト/秒)
<b>Write-KB/s</b>	書き込みのデータ転送速度 (キロバイト/秒)
<b>Transfers</b>	読み取りおよび書き込みの転送回数
<b>Disks</b>	ディスク数。
<b>Adapter-Type</b>	アダプターのタイプ。

## 共用イーサネット・アダプター

このビューには、仮想入出力サーバー (VIOS) 内の共用イーサネット・アダプター統計が表示されます。このビューを表示するには、**O** キーを押します。以下のメトリックがこのビューに表示されます。

<b>Number</b>	シリアル番号。
<b>Name</b>	共用イーサネット・アダプターの名前。
<b>Recv-KB/s</b>	受信されたデータのデータ転送速度 (キロバイト/秒)
<b>Trans-KB/s</b>	送信されたデータのデータ転送速度 (キロバイト/秒)
<b>Packin</b>	この間隔の中での受信パケット数/秒。
<b>Packout</b>	この間隔の中での送信パケット数/秒。
<b>Insize</b>	この間隔の中で着信したパケットの平均サイズ/秒。
<b>Outsize</b>	この間隔の中で発信したパケットの平均サイズ/秒。

## 詳細検査 OK/警告/危険

このビューには、プロセッサ、メモリー、およびディスクの統計情報が表示されます。また、このビューには、事前定義のしきい値を超えるシステム・メトリックに基づいて、OK、警告、危険などの状況メッセージも表示されます。このビューを表示するには、**v** キーを押します。

## 詳細ページ統計

このビューにはページ統計が表示されます。このビューを表示するには、**M** キーを押します。

**M** キーを 1 回押すと、このビューには統計情報が含まれます (ページ数単位)。 **M** キーを 2 回押すと、ページ統計情報が表示されます (メガバイト単位)。

以下のメトリックがこのビューに表示されます。

<b>Numframes</b>	このページ・サイズの実メモリ・フレーム数。
<b>Numfrb</b>	フリー・リスト上のページ数。
<b>Numclient</b>	クライアント・フレーム数。
<b>Numcompress</b>	圧縮セグメント内のフレーム数。
<b>Numperm</b>	非作業用セグメント内のフレーム数。
<b>Numvpages</b>	アクセスされた仮想ページ数。
<b>Minfree</b>	最小フリー・リスト。
<b>Maxfree</b>	最大フリー・リスト。
<b>Numpout</b>	ページ・アウト回数。
<b>Numremote</b>	リモート・ページ・アウト回数。
<b>Numwseguse</b>	作業用セグメントとして使用中のページ数。
<b>Numpseguse</b>	永続セグメントとして使用中のページ数。
<b>Numclseguse</b>	クライアント・セグメントとして使用中のページ数。
<b>Numwsegin</b>	作業用セグメントとしてピンされたページ数。
<b>Numpsegin</b>	永続セグメントとしてピンされたページ数。
<b>Numclsegin</b>	クライアント・セグメントとしてピンされたページ数。
<b>numpgsp_pgs</b>	割り当てられたページ・スペース数。
<b>numralloc</b>	リモート割り当て数。
<b>pfrsvdblks</b>	システム予約済みブロック数。
<b>Pfavail</b>	ピン用に使用可能なページ数。
<b>Pfpinavail</b>	ピン用に使用可能なアプリケーション・レベル番号ページ数。
<b>system_pgs</b>	<b>V_SYSTEM</b> でマーク付けされたセグメント制御ブロック (SCB) 上のページ数。
<b>nonsys_pgs</b>	<b>V_SYSTEM</b> でマーク付けされていないセグメント制御ブロック (SCB) 上のページ数。
<b>Numpermio</b>	非作業用記憶域内のページ・アウト数。
<b>Pgexct</b>	ページ・フォールト数。
<b>Pgrclm</b>	ページ再利用数。
<b>Pageins</b>	ページインされたページ数。
<b>Pageouts</b>	ページアウトされたページ数。
<b>Pgspgins</b>	ページ・スペースからページインされたページ数。
<b>Pgspgouts</b>	ページ・スペースからページアウトされたページ数。
<b>Numsios</b>	開始された入出力回数。
<b>Numiodone</b>	完了した入出力回数。
<b>Zerofills</b>	ゼロで埋め込まれたページ数。
<b>Exfills</b>	exec で埋め込まれたページ数。
<b>Scans</b>	クロックによるページ・スキャン回数。
<b>Cycles</b>	クロック・ハンド・サイクル数。
<b>pgsteals</b>	ページ・スチール数。

## ファイバー・チャネル・アダプター統計

このビューには、ファイバー・チャネル・アダプターに関する情報が表示されます。このビューを表示するには、脱字記号 (^) キーを押します。このビューには以下のメトリックが含まれています。

<b>Number</b>	シリアル番号。
<b>Name</b>	ファイバー・チャネル・アダプターの名前。
<b>Receive-KB/s</b>	受信されたデータのデータ転送速度 (キロバイト/秒)
<b>Transmit-KB/s</b>	送信されたデータのデータ転送速度 (キロバイト/秒)

Requests In	この間隔の中での受信要求数/秒。
Requests Out	この間隔の中での送信要求数/秒。
Outsize	この間隔の中での出力パケット平均サイズ/秒。

## 環境変数

環境変数 **NMON\_START**、**NMON\_END**、**NMON\_SNAP**、および **NMON\_ONE\_IN** が、`nmon` フォーマットで記録時に外部データを収集するのに使用されます。

### **NMONCMD0, NMONCMD1, ..., NMONCMD63**

これらの環境変数が設定されると、これらの変数の中に設定されたプロセスだけをモニターできます。別の方法として、**-C** フラグを使用して、`nmon` コマンドのプロセス・リスト中でコマンドを制限できます。例えば、`nmon -C db2:websm:nmon:topas` コマンドを実行できます。`nmon` コマンド開始時に表示される初期セットのパネルに対応した、キー・ストロークのセットを含みます。以下の値に合わせて、**TIMESTAMP** 変数を指定できます。

### **NMON**

### **TIMESTAMP**

#### **TIMESTAMP = 0**

記録される行には、その行の先頭で `nmon Tnnnn` タイム・スタンプが含まれ、`nmon` データ・ファイルを処理します。

#### **TIMESTAMP = 1**

この行には、時、分、秒、日、月、年を持ったタイム・スタンプが含まれます。分析用の `nmon` ファイルとデータをマージしたくない場合に、この値を使用できます。

### **NMON\_START**

### **NMON\_END**

### **NMON\_SNAP**

`nmon` 記録開始時に起動される外部コマンド。  
`nmon` 記録終了時に起動される外部コマンド。  
 メトリックを定期的に記録するのに起動される外部コマンド。  
 以下の値に合わせて、**NMON\_ONE\_IN** 変数を指定できます。

### **NMON\_ONE\_IN**

#### **NMON\_ONE\_IN=1**

記録が行われるたびに `snap` コマンドを実行します。

#### **NMON\_ONE\_IN=n**

*n* パラメーターで指定された記録回数が処理された後に、`snap` コマンドを実行します。

## 例

- 30 秒ごとにデータを収集して、現行ディレクトリーに 2 時間記録する `nmon` を生成するには、以下のコマンドを入力します。

```
nmon -f -s 30 -c 240
```

- `nmon` コマンドの開始直後にメモリーとプロセッサの統計を表示するには、以下のステップを行います。

- 次のコマンドを入力します。

```
export NMON=mc
```



- b. **nmon** コマンドを実行します。
3. 10 秒で画面をリフレッシュしながら、**nmon** コマンドを 20 秒間実行するには、以下のコマンドを入力します。

```
nmon -c 10 -s 2
```

4. モノクロ・モードで **nmon** を実行するには、以下のコマンドを入力します。

```
nmon -b
```

5. プロセス情報を表示するには、以下のステップを行います。

a. **nmon** コマンドを実行します。

b. **t** キーを押します。

6. **nmon** 提供のビュー・リストを表示するには、**h** キーを押します。

7. 以下の例では、外部データを収集するためのステップが説明されています。この例では、**mystart** ファイル、**mysnap** ファイル、および **myend** ファイルが実行可能であり、**\$PATH** で定義されたパスにあります。

- a. 以下に示したとおりに、環境変数を設定します。

```
$export TIMESTAMP=0
$export NMON_START="mystart"
$export NMON_SNAP="mysnap"
$export NMON_END="myend"
$export NMON_ONE_IN=1
```

上記の例で、1 の値が **NMON\_ONE\_IN** 環境変数に対するデフォルト値です。この値では、**nmon** 記録の各スナップショットごとに、外部的に記録されるデータを 1 セット生成します。

- b. 以下のようにして、**mystart** ファイルの内容を変更します。

```
ps -ef >start_ps.txt
echo "PROCCOUNT,Process Count, Procs" >ps.csv
```

- c. 以下のようにして、**mysnap** ファイルの内容を変更します。

```
echo PROCCOUNT,$1,`ps -ef | wc -l` >>ps.csv
```

- d. 以下のようにして、**myend** ファイルの内容を変更します。

```
echo PROCCOUNT,$1,`ps -ef | wc -l` >>ps.csv
```

- e. 以下のようにして、**nmon** コマンドを実行します。

```
nmon -f -s 2 -c 10
```

20 秒で記録が終了します。

**ps.csv** ファイルの出力は、次の例のようになります。

```
PROCCOUNT,Process Count, Procs
PROCCOUNT,T0001, 43
PROCCOUNT,T0002, 43
PROCCOUNT,T0003, 43
PROCCOUNT,T0004, 43
PROCCOUNT,T0005, 43
PROCCOUNT,T0006, 43
PROCCOUNT,T0007, 43
PROCCOUNT,T0008, 43
PROCCOUNT,T0009, 44
PROCCOUNT,T0010, 44
PROCCOUNT,T0010, 44
```

外部記録処理により生成された **ps.csv** ファイルと、生成された **nmon** ファイルを連結するには、以下のコマンドを入力します。



```
cat filename.nmon ps.csv > c.csv
```

グラフを入手するには、nmon アナライザーで c.csv ファイルをオープンします。

## 位置

/usr/bin/nmon

/usr/bin/topasrec

## 関連情報

topas コマンド。

topas/topasout 用の SMIT パネル

---

## no コマンド

### 目的

ネットワーク・チューニング・パラメーターを管理します。

### 構文

```
no [ -p | -r ] { -o Tunable[=NewValue] }
```

```
no [ -p | -r ] { -d Tunable }
```

```
no [ -p | -r ] { -D }
```

```
no [ -p | -r ] -a
```

```
no -h [ Tunable ]
```

```
no -L [ Tunable ]
```

```
no -x [ Tunable ]
```

注: 複数のフラグ **-o**、**-d**、**-x**、および **-L** を指定できます。

### 説明

**no** コマンドは、ネットワーク・チューニング・パラメーターを構成するために使用します。 **no** コマンドは、ネットワーク・チューニング・パラメーターの現行値または次のブート値を設定または表示します。このコマンドは、永久的な変更を行うことも、次のリブートまで変更を遅らせることもできます。このコマンドでパラメーターを設定するか、表示するかは、指定するフラグによって決まります。 **-o** フラグを指定すると、設定および表示を行います。これは、パラメーターの値を表示するかまたはパラメーターに新しい値を設定するかのいずれかを行うことができます。 **no** コマンドを使用してネットワーク・オプションを変更すると、LOG\_KERN 機能を使用してメッセージを syslog に記録します。ネットワーク・パラメーターがお互いにどのようなやり取りをするかについての詳細は、ネットワークおよびコミュニケーションのマネージを参照してください。

## チューナブル・パラメーターの変更の影響を理解する

このコマンドの使用には注意してください。no コマンドを正しく使用しないと、システムが操作不能になることがあります。

チューナブル・パラメーターを変更する前に、その目的を完全に理解するために、まず以下の「チューナブル・パラメーター」セクションで、チューナブル・パラメーターの特性のすべてについて注意して読み、「参照」ポインターがあればこれに従ってください。チューナブル・パラメーターについて詳しくは、以下の 263 ページの『ネットワーク・チューナブル・パラメーター』および 306 ページの『ストリーム・チューナブル・パラメーター』を参照してください。

その後、このパラメーターの「診断」セクションおよび「チューニング」セクションが本当にご使用の状態に適用されるか、およびこのパラメーターの値の変更がシステムのパフォーマンスを改善する援助となるか、確認する必要があります。

診断セクションとチューニング・セクションの両方に「N/A」とのみ記されている場合は、AIX の開発側から特に指示がないかぎり、多くの場合、このパラメーターを変更する必要はありません。

## フラグ

- a** すべてのチューナブル・パラメーターの現行値、レポート値 (**-r** を指定した場合) または永久値 (**-p** を指定した場合) を、*Tunable = Value* の対にして、1 行に 1 つずつ表示します。永久オプションの場合、パラメーターの値はそのレポート値と現行値が等しい場合のみ表示されます。そうでない場合は値として NONE が表示されます。
- d *Tunable*** *Tunable* をデフォルト値にリセットします。 *Tunable* を変更する必要がある場合 (つまり、現在、デフォルト値に設定されていない) で、タイプが **Bosboot** または **Reboot** であるか、またはタイプが **Incremental** でデフォルト値が変更されており、**-r** が指定されていない場合、変更は行われず、警告が表示されます。
- D** すべての *tunable* をデフォルト値にリセットします。変更する必要がある *tunable* がタイプ **Bosboot** または **Reboot** であるか、またはタイプが **Incremental** でデフォルト値が変更され、**-p** も **-r** も組み合わせで指定されていない場合、変更は行われず、警告が表示されます。
- h [*Tunable*]** *Tunable* パラメーターのヘルプを表示します (このパラメーターが指定されている場合)。このパラメーターが指定されていない場合は、**no** コマンド使用ステートメントを表示します。
- L [*Tunable*]** 1 つまたはすべての *Tunable* の特性を、次の形式で 1 行に 1 つずつリストします。

```
NAME          CUR  DEF  BOOT  MIN  MAX  UNIT  TYPE
DEPENDENCIES
-----
General Network Parameters
sockthresh    85  85   85   0   100  %_of_thewall  D
fasttimo      200 200  200  50  200  millisecond    D
inet_stack_size 16  16   16   1   kbyte          R
...
where:
CUR = current value
DEF = default value
BOOT = reboot value
MIN = minimal value
MAX = maximum value
UNIT = tunable unit of measure
TYPE = parameter type: D (for Dynamic), S (for Static), R (for Reboot),
      B (for Bosboot), M (for Mount), I (for Incremental), C (for Connect), and d (for Deprecated)
DEPENDENCIES = list of dependent tunable parameters, one per line
```

**-o** *Tunable* [=NewValue ] *Tunable* の値を表示するかまたは *NewValue* の値に設定します。パラメーターを変更する必要があり (指定された値が現行値と異なる)、このタイプが *Bosboot* または *Reboot* であるか、またはタイプが *Incremental* でその現行値が指定された値より大きく、かつ **-r** が指定されていない場合、変更は行なわれず、警告が表示されます。

新しい値を指定せずに **-r** を使用すると、*Tunable* の次のブート値が表示されます。新しい値を指定せずに **-p** を使用すると、*tunable* の現行値と次のブート値が同じ場合のみ値が表示され、そうでない場合は、値 *NONE* が表示されます。

**-p** **-o**、**-d** または **-D** を指定した場合、現行値とリブート値の両方を変更します。つまり、現行値の更新だけでなく、*/etc/tunables/nextboot* ファイルの更新をオンにします。 *Reboot* および *Bosboot* タイプのパラメーターでは現行値を変更できないため、これらの組み合わせは使用できません。

新しい値を指定せずに **-a** または **-o** を使用した場合、パラメーターの現行値と次のブート値が同じ場合のみ、値が表示されます。そうでない場合は値として *NONE* が表示されます。

**-r** **-o**、**-d** または **-D** を指定した場合、リブート値を変更します。つまり、*/etc/tunables/nextboot* ファイルの更新をオンにします。タイプ *Bosboot* のパラメーターを変更すると、*bosboot* を実行するというプロンプトがユーザーに出されます。新しい値を指定せずに **-a** または **-o** を使用すると、パラメーターの現行値ではなく次のブート値が表示されます。

**-x** [*Tunable*] 1 つまたはすべての *Tunable* の特性を、次のスプレッドシート形式で 1 行に 1 つずつリストします。

```
tunable,current,default,reboot,min,max,unit,type,{dtunable }
```

where:

```
current = current value
default = default value
reboot = reboot value
min = minimal value
max = maximum value
unit = tunable unit of measure
type = parameter type: D (for Dynamic), S (for Static), R (for Reboot),
      B (for Bosboot), M (for Mount), I (for Incremental),
      C (for Connect), and d (for Deprecated)
dtunable = space separated list of dependent tunable parameters
```

タイプ *Mount* のパラメーターを変更すると (**-o**、**-d**、または **-D** フラグを指定)、その変更は将来のマウントについてのみ有効であるという警告メッセージが表示されます。

タイプ *Connect* のパラメーターを (**-o**、**-d**、または **-D** を使用して、) 変更すると、*inetd* が再始動され、この変更は将来のソケット接続に対してのみ有効であることを示す警告メッセージが表示されます。

**-r** を指定せずに、タイプ *Bosboot* または *Reboot* のパラメーターを (**-o**、**-d**、または **-D** を使用して) 変更すると、エラー・メッセージが表示されます。

タイプ *Incremental* のパラメーターの現在の値を、現在の値よりも小さい新規の値により、(**-o**、**-d**、または **-D** を指定し、**-r** を指定しないで) 変更すると、エラー・メッセージが表示されます。

## チューナブル・パラメーターのタイプ

チューニング・コマンド (*no*、*nfso*、*vmo*、*ioo*、*schedo*、および *raso*) で取り扱われるすべてのチューナブル・パラメーターは、下記のカテゴリに分類されています。

*Dynamic*                      パラメーターをいつでも変更できる場合

Static	パラメーターをいかなる時にでも変更できない場合
Reboot	パラメーターをリブート時にのみ変更できる場合
Bosboot	bosboot を実行してマシンをリブートする場合にのみパラメーターを変更できる場合
Mount	パラメーターの変更が将来のファイルシステムまたはディレクトリーのマウントにのみ有効である場合
Incremental	ブート時を除き、パラメーターを増やすことだけが可能な場合
Connect	パラメーターの変更が将来のソケット接続にのみ有効である場合
Deprecated	このパラメーターの変更が AIX の現行リリースでサポートされなくなっている場合

タイプ **Bosboot** のパラメーターの場合は、変更が加えられるたびに、チューニング・コマンドは自動的に、**bosboot** コマンドを実行したいかどうかを尋ねるプロンプトを出します。タイプ **Connect** のパラメーターの場合は、**pre520tune** が使用不可にされていると、チューニング・コマンドは自動的に **inetd** デーモンを再始動します。

**no** コマンドにより管理されるパラメーターの現行セットには、**Reboot**、**Static**、**Dynamic**、**Incremental**、および **Connect** のタイプがあります。

## 互換モード

5.2 より前の互換モードで実行する場合 (**sys0** の **pre520tune** 属性で制御される。AIX 5.2 AIX 5.2 互換モードを参照)、リポート値は、タイプ **Bosboot** のパラメーターの場合を除き、実際には意味がありません。このモードでは、ブート時にリポート値は適用されないからです。

5.2 より前の互換モードでは、チューニング・パラメーターへのリポート値の設定は、ブート中に呼び出されるスクリプト内でチューニング・コマンドを呼び出すことにより行います。したがって、タイプ **Reboot** のパラメーターは **-r** フラグなしで設定できるので、既存のスクリプトは従来どおり作動します。

このモードは、マシンを AIX 5L バージョン 5.2 に移行すると、自動的にオンになります。完全なインストールの場合、これは OFF になり、パラメーターのリポート値は、リポート中に **/etc/tunables/nextboot** ファイルの内容を適用することにより設定されます。このモードの場合のみ、**-r** および **-p** フラグは完全に機能します。詳細については、*AIX 5L Version 5.3 Performance Tools Guide and Reference* のカーネル・チューニングのセクションを参照してください。

## ネットワーク・チューナブル・パラメーター

<b>arpqsize</b>	<p><b>目的:</b> ARP 応答を待っている間にキューに加えられるパケットの最大数を指定します。デフォルト値は 1 です。</p> <p><b>値:</b></p> <p>デフォルト: 12</p> <p>範囲: 1 から MAXSHORT</p> <p>タイプ: Dynamic</p> <p><b>診断:</b> 該当しない</p> <p><b>チューニング:</b></p> <p>このパラメーターは、イーサネット、802.3、トークンリングおよび FDDI インターフェースでサポートされます。このパラメーターは、AIX バージョン 4.1.5、AIX バージョン 4.2.1 およびそれ以降に適用されます。</p> <p><b>参照:</b> ARP cache tuning のセクション。</p>
-----------------	---

## arpt\_killc

**目的:** 完全な ARP エントリーを削除するまでの時間 (分) を指定します。

**値:**

デフォルト: 20

範囲: 0 から 255

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

安定したネットワークで ARP 活動を減らすには、**arpt\_killc** を増やします。

**参照:** ARP cache tuning のセクション。

## arptab\_bsiz

**目的:** アドレス解決プロトコル (ARP) テーブル・バケット・サイズを指定します。

**値:**

デフォルト: 7

範囲: 1 から MAXSHORT

タイプ: Reboot

**診断:** **netstat -p arp** は、送信された ARP パケットの数および ARP テーブルから消去された ARP エントリーの数を表示します。多くのエントリーが消去される場合は、ARP テーブル・サイズを大きくする必要があります。ARP テーブル・ハッシュ分布を表示するには、**arp -a** を使用します。

**チューニング:**

該当しない

**参照:** ARP cache tuning のセクション。

## arptab\_nb

**目的:** ARP テーブル・バケット数を指定します。

**値:**

デフォルト: 149

範囲: 1 から MAXSHORT

タイプ: Reboot

**診断:** **netstat -p arp** は、送信された ARP パケットの数および ARP テーブルから消去された ARP エントリーの数を表示します。多くのエントリーが消去される場合は、ARP テーブル・サイズを大きくする必要があります。ARP テーブル・ハッシュ分布を表示するには、**arp -a** を使用します。

**チューニング:**

多数のクライアントやサーバーを持つシステムの場合、この値を増やします。デフォルトでは  $149 \times 7 = 1043$  ARP エントリーを提供していますが、これは均一なハッシュ分布を想定しています。

**参照:** ARP cache tuning のセクション。

## bcastping

**目的:** ブロードキャスト・アドレスへの ICMP エコー・パケットに応答できるようにします。

**値:**

デフォルト: 0 (オフ)

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

デフォルトは、ブロードキャスト・アドレスへのエコー・パケットへ応答しないことです。これによって、複数のマシンが 1 つのブロードキャスト・アドレスに対して応答したときに発生する可能性がある、ネットワーク上の「ブロードキャスト・ストーム」を防止できます。

## clean\_partial\_conns

**目的:** SYN (シーケンス番号を同期化する) アタックを避けるかどうかを指定します。

**値:**

デフォルト: 0 (オフ)

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

このオプションは、ネットワーク・アタックから保護する必要のあるサーバーについては、オンにすべきです。オンにすると、アタック以外の新しい接続用の余裕を作るため、部分的な接続をランダムに除去します。

## delayack

**目的:** ある一定の TCP パケットの ACK を遅らせ、代わりに、送信される次のパケットと ACK を抱き合わせて送信を試みます。

**値:**

デフォルト: 0

範囲: 0 から 3

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

このアクションは、宛先ポートが `delayackports` パラメーターのリストに指定されている接続にのみ実行されます。これを使用すると、送信されるパケットの総数が減るので、HTTP サーバーとの通信時にパフォーマンスが向上します。パラメーターには、次の 4 つの値のいずれかを指定できます。

**0** 遅らせない。通常操作。

**1** サーバーの SYN (シーケンス番号の同期化) の ACK を遅らせる。

**2** サーバーの FIN (送信側のバイト・ストリームが終了した) の ACK を遅らせる。

**3** SYN と FIN の両方の ACK を遅らせる。

## delayackports

**目的:** delayack port オプションで定義した操作が行われる出力先ポートのリストを指定します。この属性では、最大 10 個までのポートのリストを指定します。

**値:**

デフォルト: {}

範囲: ポート番号のリスト (最大 10)

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

パラメーターは 10 個までのポートのリストであり、各ポートをコンマで区切り、中括弧で囲みます。以下に例を示します。

```
no -o delayackports={80,30080}
```

リストをクリアするには、オプションを {} にします。

## dgd\_packets\_lost

**目的:** デッド・ゲートウェイ検出がゲートウェイをダウン状態と判定するまでに連続して失われる必要のあるパケット数を指定します。

**値:**

デフォルト: 3

範囲: 1 から MAXSHORT

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## dgd\_ping\_time

**目的:** アクティブ・デッド・ゲートウェイ検出による、1 つのゲートウェイの複数の ping 間で経過する必要がある秒数を指定します。

**値:**

デフォルト: 5

範囲: 1 から MAXLONG

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**



## **dgd\_retry\_time**

**目的:** 経路のコストがバッド・デッド・ゲートウェイ検出によって格上げされたときに、何分間格上げされたままになる必要があるのかを指定します。この時間が経過すると、その経路のコストはユーザーが構成した値に復元されます。

**値:**

デフォルト: 5

範囲: 1 から MAXSHORT

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## **directed\_broadcast**

**目的:** ゲートウェイにブロードキャストを送ることを許すかどうかを指定します。

**値:**

デフォルト: 0 (オフ)

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

値 1 を指定すると、ゲートウェイに送られるパケットを、そのゲートウェイの他方の端にあるネットワーク上でブロードキャストできるようになります。

## **extendednetstats**

**目的:** ネットワーク・メモリー・サービスに関してより拡張された統計情報を使用可能にします。

**値:**

デフォルト: 0 (オフ)

範囲: 0 または 1

タイプ: Reboot

**診断:** 該当しない

**チューニング:**

このパラメーターのデフォルトは 0 であり、オフを示します。その理由は、より多くの統計を取るにより SMP システム上のシステム・パフォーマンスが低下するためです。

**fasttimo**

**目的:** TCP 高速タイムアウト・タイマーのミリ秒の遅延を設定できるようにします。このタイムアウトは、遅らせた ACK を送信するために、システムが TCP 制御ブロックをスキャンする頻度を制御します。

**値:**

デフォルト: 200

範囲: 50 から 200 ミリ秒

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

このタイマーの値を小さくすると、IBM 以外の一部のシステムでパフォーマンスが改善されることがあります。しかし、その結果システム使用率もわずかに増加することがあります。

**icmp6\_errmsg\_rate**

**目的:** 1 秒単位で送信可能な ICMP v6 エラー・メッセージの数の上限を指定します。これによって、ICMP v6 エラー・メッセージが使用している帯域幅を除外しないようにします。

**値:**

デフォルト: 10 msg/秒

範囲: 1 から 255

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

**icmpaddressmask**

**目的:** システムが ICMP アドレス・マスク要求に応答するかどうかを指定します。

**値:**

デフォルト: 0 (オフ)

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

デフォルト値 0 を設定すると、ネットワークは受信するすべての ICMP アドレス・マスク要求を、何も通知せずに無視します。

**ie5\_old\_multicast\_mapping**

**目的:** 値 1 の場合、トークンリングの IP マルチキャストを、機能アドレスではなくブロードキャスト・アドレスにマップすることを指定します。

**値:**

デフォルト: 0 (オフ)

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## ifsize

**目的:** 1 つのタイプ (例えば、イーサネット) のインターフェース当たりのネットワーク・インターフェース構造の最大数を指定します。イーサネット・インターフェース構造の場合は、いくつのイーサネット・インターフェース構造でも扱えるようインフラストラクチャーが動的に拡張するため、この制限は適用されません。

**値:**

デフォルト: 256

範囲: 8 から 1024

タイプ: Reboot

**診断:** 該当しない

**チューニング:**

ホット・プラグ・アダプターをサポートするマシン上および DLPAR 構成上では、**ifsize** の値を大きくする必要があります。これは、アダプターをフライ上に追加でき、静的インターフェース・テーブルがこのシステムまたは区画に追加できるアダプターの最悪の場合の数を受け入れるのに十分な大きさでなければならないためです。システムは、ブート時に 1 つのタイプのアダプターが **ifsize** の現行値で許容されている数より多く存在することを検出した場合、現存するアダプターの数をサポートするために値を自動的に増やします。

**参照:** 該当しない

## inet\_stack\_size

**目的:** inet 割り込みスタック・テーブルのサイズをキロバイトで指定します。

**値:**

デフォルト: 16

範囲: 1 から MAXSHORT

タイプ: Reboot

**診断:** 該当しない

**チューニング:**

これは、最適化されていないデバッグ・カーネルまたは netinet カーネル拡張を使用して実行している場合に必要です。これは、割り込み上に置かれるため、ピン続行スタック・コード (構成不可能) とは異なります。このパラメーターを変更する必要があるのは、割り込みスタック・オーバーフローにより、システムがパニック状態になる場合のみです。

## ip\_ifdelete\_notify

**目的:** インターフェース・アドレスが削除されたときに、既存の TCP 接続に通知すべきかどうかを指定します。このオプションは、次のファイルセット・レベルから使用可能です。

AIX 5L V5.1: bos.net.tcp.client 5.1.0.65

AIX 5L V5.2: bos.net.tcp.client 5.2.0.75

AIX 5L V5.3: bos.net.tcp.client 5.3.0.30

**値:**

デフォルト: 0 (オフ)

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

インターフェース・アドレスが削除されるときに、削除されているインターフェース・アドレスにローカルにバインドされたすべての既存の TCP 接続に、エラー **ENETDOWN** によって通知する必要があることを指定します。

**注:**

1. 既存の FTP/Telnet 接続は、**ENETDOWN** エラーが戻されると切断されます。
2. 次のファイルセット・レベルまで、このオプションのデフォルト値は 1 です。

AIX 5L V5.1: bos.net.tcp.client 5.1.0.70

AIX 5L V5.2: bos.net.tcp.client 5.2.0.97

AIX 5L V5.3: bos.net.tcp.client 5.3.0.52

## ip\_nfrag

**目的:** IP フラグメント化アタックを回避するかどうかを指定します。

**値:**

デフォルト: 200

範囲: 1 から MAXSHORT

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

一度に IP 再アセンブリー・キューに保持可能な、IP パケットのフラグメントの最大数を指定します。デフォルト値の 200 は、IP 再アセンブリー・キューに IP パケットのフラグメントを 200 まで保持できます。

## ipforwarding

**目的:** カーネルがパケットを転送するかどうかを指定します。

**値:**

デフォルト: 0 (オフ)

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

システムが IP ルーターとして働く場合は、このパラメーターを 1 にします。

**参照:** traceroute コマンド

## ipfragttl

**目的:** IP フラグメントの存続時間を 0.5 秒単位で指定します。

**値:**

デフォルト: 2

範囲: 1 から 255

タイプ: Dynamic

**診断:** フラグメントがタイムアウトの後でドロップされました (**netstat -p ip**)。

**チューニング:**

タイムアウトの後でドロップされた IP フラグメントの値がゼロ以外の場合、**ipfragttl** を増やすと再転送が減ります。

## ipignoreredirects

**目的:** 受信したりダイレクトを処理するかどうかを指定します。

**値:**

デフォルト: 0 (リダイレクトする)

範囲: 0 または 1 (1 はリダイレクトを無視する)

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

このオプションは、AIX バージョン 4.2.1 以降にのみ適用されます。

## ipqmaxlen

**目的:** IP プロトコル入力キューに入れることができる受信パケットの数を指定します。

**値:**

デフォルト: 100

範囲: 100 から MAXINT

タイプ: Reboot

**診断:** **ipintrq** オーバーフローを検査するか (**netstat -s**)、またはクラッシュを使用して IP 入力キューのオーバーフロー・カウンターを見ます。

**チューニング:**

システムが多数のループバック・セッションを使用している場合はサイズを増やします。ほとんどのオペレーティング・システムのネットワーク・ドライバーは、IP を直接呼び出し、IP キューを使用しません。このようなデバイスで **ipqmaxlen** を増やしても効果はありません。

**参照:** 該当しない

## ipssendredirects

**目的:** カーネルがリダイレクト・シグナルを送信するかどうかを指定します。

**値:**

デフォルト: 1 (リダイレクトを送信する)

範囲: 0 (リダイレクトを送信しない) または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

これは、パフォーマンスに関する構成上の決定です。

## ipsrcrouteforward

**目的:** システムがソース経路指定パケットを転送するかどうかを指定します。

**値:**

デフォルト: 1 (オン)

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

デフォルト値が 1 の場合、ソース経路指定パケットは転送されます。値が 0 の場合、宛先がないすべてのソース経路指定パケットは廃棄されます。このパラメーターは、AIX 4.2.1 以降にのみ適用されます。

## ipsrcrouterrecv

**目的:** システムがソース経路指定パケットを受け入れるかどうかを指定します。

**値:**

デフォルト: 0 (オフ)

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

デフォルト値 0 は、このシステムに向けられたすべてのソース経路指定パケットを廃棄します。値 1 の場合、ソース経路指定パケットを受け取ります。このパラメーターは、AIX 4.2.1 以降にのみ適用されます。

## ipsrcroutesend

**目的:** アプリケーションがソース経路指定パケットを送信できるかどうかを指定します。

**値:**

デフォルト: 1 (オン)

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

デフォルト値 1 は、ソース経路指定パケットを転送できるようにします。値 0 の場合、アプリケーションがソース経路指定オプションを設定しようとする、setsockopt() はエラーを戻し、すべてのソース経路指定オプションが発信パケットから除去されます。このパラメーターは、AIX 4.2.1 以降にのみ適用されます。

## ip6\_defttl

**目的:** 他のホップ・カウントが指定されていない場合に、IPv6 (インターネット・プロトコル・バージョン 6) パケットに使用するデフォルトのホップ・カウントを指定します。

**値:**

デフォルト: 64  
範囲: 1 から 255  
タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## ip6\_prune

**目的:** 期限切れの経路指定に関して IPv6 経路指定テーブルを検査する頻度を秒単位で指定します。

**値:**

デフォルト: 1  
範囲: 1 から MAXLONG  
タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## ip6forwarding

**目的:** カーネルが IPv6 パケットを転送するかどうかを指定します。

**値:**

デフォルト: 0 (オフ)  
範囲: 0 または 1  
タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

デフォルト値 0 は、IPv6 パケットがローカル・システムあてでない場合、転送しません。値 1 を指定すると、転送が可能になります。

## ip6srcrouteforward

**目的:** システムがソース経路指定 IPv6 パケットを転送するかどうかを指定します。

**値:**

デフォルト: 1 (オン)  
範囲: 0 または 1  
タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

デフォルト値が 1 の場合、ソース経路指定パケットは転送されます。値が 0 の場合、宛先にはないすべてのソース経路指定パケットは廃棄されます。



## llsleep\_timeout

**目的:** リンクのローカル・タイムアウトのタイムアウト値を秒単位で指定します (multi\_homed=1 の場合に使用)

**値:**

デフォルト: 3

範囲: 1 から MAXINT

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## lo\_perf

**目的:** オフレベル割り込み処理用に複数のハンドラー構造を作成することにより、ループバックでのパフォーマンスを向上します。

**値:**

デフォルト: 1 (オン)

範囲: 0 または 1

タイプ: Reboot

**診断:** 該当しない

**チューニング:**

このオプションはオフレベル・ハンドラーのエイを **ipintrqs** のエイと共に作成します。このエイには、パケットに適用されたハッシュ機能に基づいてそれらのパケットが保持されます。値 0 を指定すると、ループバック処理ルーチンで、1 つの **ipintrq** と共に 1 つのオフレベル割り込みハンドラーが使用されます。このオプションは AIX 5.2 以降にのみ適用されます。

## main\_if6

**目的:** リンク・ローカル・アドレスに使用するインターフェースを指定します。これは、初期経路をセットアップするために **autoconf6** のみで使用されます。

**値:**

デフォルト: 0

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## main\_site6

**目的:** サイト・ローカル・アドレス経路指定に使用するインターフェースを指定します。これは、**multi\_homed** が 3 に設定されている場合にのみ使用されます。

**値:**

デフォルト: 0

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## maxnip6q

**目的:** IPv6 パケット再組み立てキューの最大数を指定します。

**値:**

デフォルト: 20

範囲: 1 から MAXSHORT

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## maxttl

**目的:** RIP (ルーティング情報プロトコル) パケットの存続時間を秒単位で指定します。

**値:**

デフォルト: 255

範囲: 1 から 255

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## mpr\_policy

**目的:** Multipath Routing とともに使用されるグローバル経路指定ポリシーを指定します。選択可能な経路指定ポリシーは次のとおりです。

- 重み付けラウンドロビン (1) - (**route** コマンドを使用して) 複数経路に割り当てられたユーザー構成の重みを基にして、ラウンドロビンが適用されます。重みが構成されていない場合は、プレーンなラウンドロビンと同一に動作します。
- ランダム (2) - 経路をランダムに選択します。
- 重み付けランダム (3) - ユーザー構成の重みとランダム化ルーチンを基にして、経路を選択します。ポリシーはすべての経路の重みを合計し、0と重みの合計の間の乱数を選びます。この数字がゼロになるまで、個々の各重みが、重みの合計から除去されます。これにより、選択可能な経路の合計数の範囲内で経路が選ばれます。
- 最低使用状況 (4) - 通る現行接続の数が最も少ない経路を選びます。
- ハッシュ・ベース (5) - ハッシュ・ベースのアルゴリズムが、宛先 IP アドレスに基づいてハッシュすることにより、経路を選択します。

**値:**

デフォルト: 1  
範囲: 1 から 5  
タイプ: Dynamic

**診断:** 該当しない

**チューニング:**  
該当しない

## multi\_homed

**目的:** マルチホームの IPv6 ホスト・サポートのレベルを指定します。

**値:**

デフォルト: 1  
範囲: 0 から 3  
タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

これは、宛先ポートが `delayackports` パラメーターのリストに指定されている接続にのみ実行されます。HTTP サーバーとの通信時に、これを使用してパフォーマンスを向上することができます。パラメーターには、次の 4 つの値のいずれかを指定できます。

**0** - AIX バージョン 4.3 の元の機能を指定します。

**1** - リンク・ローカル・アドレスは、そのリンク・ローカル・アドレスのそれぞれのインターフェースに照会することで解決されます。

**2** - リンク・ローカル・アドレスは、`main_if6` で定義されたインターフェースのみ検査されます。

**3** - リンク・ローカル・アドレスは `main_if6` で定義されたインターフェースのみ検査され、サイト・ローカル・アドレスは `main_site6` インターフェースにのみ経路指定されます。

## nbcb\_limit

**目的:** ネットワーク・バッファ・キャッシュ (NBC) に使用できるメモリの最大合計量を K バイトで指定します。

**値:**

デフォルト: **thewall** から算出

範囲: 0 から MAXLONG

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

キャッシュがこの制限まで達すると、最も使用度の低いキャッシュがキャッシュからフラッシュされ、新しいオブジェクトが入るようになります。このパラメーターは、AIX 4.3.2 以降にのみ適用されます。NBC は、**send\_file()** API およびカーネル内の **get** エンジンを使用するある種の Web サーバーによってのみ使用されます。

## nbcb\_max\_cache

**目的:** ネットワーク・バッファ・キャッシュ (NBC) で許容されるキャッシュ・オブジェクトの最大サイズを、バイト単位で指定します。

**値:**

デフォルト: **nbcb\_limit** がゼロでない場合は 131072 (128K)。それ以外の場合は 0。

範囲: 1 から MAXLONG

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

このサイズより大きいデータ・オブジェクトは、NBC に入りません。このパラメーターは、AIX 4.3.2 以降にのみ適用されます。NBC は、**send\_file()** API およびカーネル内の **get** エンジンを使用するある種の Web サーバーによってのみ使用されます。

## nbcb\_min\_cache

**目的:** ネットワーク・バッファ・キャッシュ (NBC) で許容されるキャッシュ・オブジェクトの最小サイズを、バイト単位で指定します。

**値:**

デフォルト: **nbcb\_limit** がゼロでない場合は 1 バイト。それ以外の場合は 0。

範囲: 1 から 131072(128K)

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

このサイズより小さいデータ・オブジェクトは、NBC に入りません。このパラメーターは、バージョン 4.3.2 以降にのみ適用されます。NBC は、**send\_file()** API およびカーネル内の **get** エンジンを使用するある種の Web サーバーによってのみ使用されます。

## nbc\_ofile\_hashsz

**目的:** ネットワーク・バッファ・キャッシュ内のキャッシュ・オブジェクトのハッシュに使用されるハッシュ・テーブルのサイズ (スロット数) を指定します。このハッシュ・テーブル・サイズは、オープン済みファイル・エントリー、すなわち、ファイルシステムからファイルをキャッシュに入れるエントリーにのみ適用されます。この属性はハッシュ・テーブル・サイズの大きさを調整し、すべての既存のエントリーのハッシュに影響を及ぼすので、ネットワーク・バッファ・キャッシュが空の場合にのみこの属性を変更することができます。ネットワーク・バッファ・キャッシュが空でない場合は、このオプションを使用するとエラーが戻されます。

**値:**

デフォルト: 12841

範囲: 1 から 999999

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

チューニングによりハッシュ・テーブル・エントリーのより均一な配布が行われるため、ハッシュ・テーブル・サイズは通常、基本サイズとなるように選択されます。

## nbc\_pseg (AIX 4.3.3 以降)

**目的:** ネットワーク・バッファ・キャッシュ (NBC) 用に作成できる専用セグメントの最大数を指定します。

**値:**

デフォルト: 0

範囲: 0 から MAXINT

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

このオプションを 0 以外にすると、**nbc\_max\_cache** に指定されたサイズと、セグメント・サイズ (256 MB) の間のサイズを持つデータ・オブジェクトは、専用セグメント内にキャッシュされます。セグメント・サイズより大きいデータ・オブジェクトはまったくキャッシュされません。ここに指定した数の専用セグメントが NBC に存在する場合、専用セグメントの数が制限を超えないようにするために、専用セグメント内のキャッシュ・データは新しいキャッシュ・データ用にフラッシュされます。このオプションを 0 にすると、専用セグメント内のすべてのキャッシュがフラッシュされます。

## nbc\_pseg\_limit

**目的:** ネットワーク・バッファ・キャッシュ (NBC) 内の専用セグメントに許されるキャッシュ・データの最大合計サイズ (KB) を指定します。

**値:**

デフォルト: 稼働システム上の RAM の 1/4

範囲: 稼働システム上の RAM の 1/2

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

専用セグメント内にキャッシュされるデータはネットワーク・バッファ・キャッシュによりピン (固定) されるため、このオプションは、グローバル・セグメント内のネットワーク・バッファに加えて、ネットワーク・バッファ・キャッシュが使用する、固定されたメモリの量も制御できるようにします。この制限に達すると、固定されたメモリ・サイズの合計が制限を超えないようにするために、専用セグメント内のキャッシュ・データは新しいキャッシュ・データ用にフラッシュされます。このオプションを 0 にすると、専用セグメント内のすべてのキャッシュがフラッシュされます。

## ndd\_event\_name

**目的:** キャプチャーされるべき **ns\_alloc** および **ns\_free** イベントのインターフェース名のリストを指定します。これは、デバイス・ドライバのトレースを調整するデバッグ・オプション (通常は使用可能に設定されていない) です。

**値:**

デフォルト: {all}

範囲: インターフェース名のリスト (16 以内) を表す、最大 127 文字のストリング。ストリングは { で始まり、} で終わる必要があります。リスト区切り文字は : (コロン) です。

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

ストリングが {all} 以外の場合、これらの名前に関する **ns\_alloc** イベントのみがキャプチャーされます。

## ndd\_event\_tracing

**目的:** **ns\_alloc** および **ns\_free** トレース・バッファのサイズを指定します。これは、デバイス・ドライバのトレースを調整するデバッグ・オプション (通常は使用可能に設定されていない) です。

**値:**

デフォルト: 0

範囲: 0 から USHORTMAX

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

この変数の値がゼロ以外の場合、すべての **ns\_alloc** イベントおよび **ns\_free** イベントは、カーネル・バッファ内でトレースされます。このパラメーターをオンにするとパフォーマンスにマイナスの影響を与えるため、**ndd** 参照の問題を調査する場合のみ、使用可能にしてください。デフォルト値はゼロです (トレースがオフ)。 **ndd\_event\_tracing** の値が 1024 より大きい場合、同数の項目がカーネル・バッファ内でトレース用に割り当てられます。

## ndp\_mmaxtries

**目的:** 送信するマルチキャスト NDP パケットの最大数を指定します。

**値:**

デフォルト: 3

範囲: 0 から MAXLONG

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## ndp\_umaxtries

**目的:** 送信するユニキャスト NDP パケットの最大数を指定します。

**値:**

デフォルト: 3

範囲: 0 から MAXLONG

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## ndpqsize

**目的:** 隣接ディスカバリー・プロトコル (NDP) エントリーの完了時に待ち状態を保持するパケットの数を指定します (MTU パス・ディスカバリーで使用される)。

**値:**

デフォルト: 50

範囲: 1 から MAXSHORT

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない



## ndpt\_down

**目的:** 隣接ディスカバリー・プロトコル (NDP) エントリーを抑制する時間を 0.5 秒単位で指定します。このネットワーク・オプションは、AIX 5.2 以降のバージョンでは廃止になっています。

**値:**

デフォルト: 3 (1.5 秒)

範囲: 1 から MAXLONG

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## ndpt\_keep

**目的:** 隣接ディスカバリー・プロトコル (NDP) エントリーを保持する時間を 0.5 秒単位で指定します。

**値:**

デフォルト: 120 (60 秒)

範囲: 1 から MAXLONG

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## ndpt\_probe

**目的:** 最初の隣接ディスカバリー・プロトコル (NDP) プローブを送信する前に何秒待つかを示す時間を 0.5 秒単位で指定します。

**値:**

デフォルト: 5 (2.5 秒)

範囲: 1 から UINTMAX

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## ndpt\_reachable

**目的:** 隣接ディスカバリー・プロトコル (NDP) エントリーがまだ有効であるかどうかをテストする時間を、0.5 秒単位で指定します。

**値:**

デフォルト: 30 (15 秒)

範囲: 1 から UINTMAX

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## ndpt\_retrans

**目的:** 隣接ディスカバリー・プロトコル (NDP) 要求を再送する前に何秒待つかを示す時間を、0.5 秒単位で指定します。

**値:**

デフォルト: 1 (0.5 秒)  
範囲: 1 から UINTMAX  
タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## net\_buf\_size

**目的:** キャプチャーされるべき net\_malloc/free イベントのバッファ・サイズのリストを指定します。これは、**net\_malloc\_police** オプションによってどのサイズのバッファをトレースするかを制御するデバッグ・オプション (通常は使用可能に設定されていない) です。

**値:**

デフォルト: 0  
範囲: サイズのリスト (16 以内) を表す、最大 127 文字のストリング。ストリングは { で始まり、} で終わる必要があります。リスト区切り文字は : (コロン) です。  
タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

ストリングが {all} 以外の場合、これらのサイズの **net\_malloc** イベントのみがキャプチャーされます。

## net\_buf\_type

**目的:** キャプチャーされるべき net\_malloc/free イベントのバッファ・タイプを指定します。これは、**net\_malloc\_police** オプションによってどのバッファ・タイプをトレースするかを制御するデバッグ・オプション (通常は使用可能に設定されていない) です。

**値:**

デフォルト: {all}  
範囲: タイプのリストを表す、最大 127 文字のストリング。ストリングは { で始まり、} で終わる必要があります。リスト区切り文字は : (コロン) です。  
タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

ストリングが {all} 以外の場合、このタイプの **net\_malloc** イベントのみがキャプチャーされます。

## net\_malloc\_police

**目的:** **net\_malloc** および **net\_free** トレース・バッファのサイズを指定します。

**重要:** これはシステムのデバッグ・オプションであり、IBM のサービスおよびサポートによって指示があった場合のみ使用してください。このオプションをゼロ以外の値に設定すると、システムのパフォーマンスに影響します。

**値:**

デフォルト: 0

範囲: 0 から MAXLONG

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

この変数の値がゼロ以外の場合、すべての **net\_malloc** および **net\_free** バッファは、カーネル・バッファ内でシステム・トレース・フック **HKWD\_NET\_MALLOC** によりトレースされます。他のエラー検査も使用可能です。これには、空きバッファの解放、位置合わせ、バッファの上書きについての検査が含まれます。このパラメーターをオンにするとパフォーマンスが悪化するので、何らかのネットワーク問題を調査する場合のみ、使用可能にしてください。デフォルト値はゼロです (ポリシーの処理がオフ)。 **net\_malloc\_police** の値が 1024 より大きい場合、それだけの数の項目がカーネル・バッファ内でトレース用に割り当てられます。

## nonlocsrcroute

**目的:** 厳重にソースに経路指定されたパケットはローカル・ネットワーク外部のホストにアドレスする可能性があることをインターネット・プロトコルに通知します。

**値:**

デフォルト: 0

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

デフォルト値が 0 の場合は、外部ホストへ送信しません。値が 1 の場合、パケットは外部ホストあてとなります。厳重でないソース経路指定パケットは、このパラメーターの影響を受けません。これは、わずかなパフォーマンス向上を求める構成上の決定です。

## passive\_dgd

**目的:** パッシブ・デッド・ゲートウェイ検出を使用可能にするかどうかを指定します。値が 0 の場合はオフになり、値が 1 の場合はすべてのゲートウェイを使用中にすることができます。

**値:**

デフォルト: 0

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## pmtu\_default\_age

**目的:** UDP パスのパス MTU 値で、より低い値を調べるまでの、デフォルトの時間を (分単位で) 指定します。

**値:**

デフォルト: 10  
範囲: 0 から MAXSHORT  
タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

値が 0 の場合、エージングは行われません。デフォルト値は 10 分です。**pmtu\_default\_age** 値は、UDP アプリケーションでオーバーライドされます。**pmtu\_default\_age** は実行時属性です。AIX 5.3 では、UDP アプリケーションがパス MTU の減少を検出するように **IP\_DONTFRAG** ソケット・オプションを設定する必要があるため、このオプションは使用されません。

## pmtu\_expire

**目的:** 使用されないパス MTU エントリーが削除されるまでのデフォルトの時間を (分単位で) 指定します。

**値:**

デフォルト: 10  
範囲: 0 から MAXSHORT  
タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

値 0 は、PMTU エントリーが期限切れにならないことを意味します。デフォルト値は 10 分です。この属性は、AIX 5.3 以降にのみ適用されます。**pmtu\_expire** は実行時属性です。

## pmtu\_rediscover\_interval

**目的:** UDP パスおよび TCP パスのパス MTU 値で、より高い値を調べるまでの、デフォルトの時間を (分単位で) 指定します。

**値:**

デフォルト: 30  
範囲: 0 から MAXSHORT  
タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

値が 0 の場合、パス MTU 再ディスカバリーは行われません。デフォルト値は 30 分です。

## rfc1122addrchk

**目的:** RFC1122 (インターネット・ホスト通信レイヤーの要件) で指定されたアドレス妥当性検査を行いません。

**値:**

デフォルト: 0 (オフ)

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

デフォルト値 0 の場合は、アドレス妥当性検査を行いません。値が 1 の場合、妥当性検査を行います。

## rfc1323

**目的:** RFC 1323 (ハイパフォーマンス用 TCP 拡張) で指定されたウィンドウ・スケーリングおよびタイム・スタンプを使用可能にします。ウィンドウ・スケーリングにより、TCP ウィンドウ・サイズ (**tcp\_recvspace** および **tcp\_sendspace**) を 64KB (65536) より大きくすることができます。ウィンドウ・スケーリングは通常、大規模な MTU ネットワークに使用されません。

**値:**

デフォルト: 0 (オフ)

範囲: 0 または 1

タイプ: Connect

**診断:** 該当しない

**チューニング:**

デフォルト値 0 は、システム全体での RFC 拡張を使用不可にします。値が 1 の場合、すべての TCP 接続が RFC 拡張との交渉を試みることを指定します。SOCKETS アプリケーションは、**setsockopt** サブルーチンを使用して、個々の TCP 接続のデフォルトの振る舞いを変更することができます。**tcp\_sendspace** および **tcp\_recvspace** を 64 KB より大きくする前に変更を行なってください。

**参照:** TCP workload tuning のセクション。

## rfc2414

**目的:** RFC 2414 で記述して、TCP の初期ウィンドウの増加を可能にします。これがオンの場合、初期ウィンドウは調整可能な **tcp\_init\_window** の設定値に依存します。

**値:**

デフォルト: 1 (オン)

範囲: 0 または 1 (オン)

タイプ: Connect

**診断:** 該当しない

**チューニング:**

該当しない

## route\_expire

**目的:** クローン作成による未使用の経路、またはリダイレクトによって作成および変更された未使用の経路の有効期限が切れるかどうかを指定します。

**値:**

デフォルト: 1 (オン)

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

値 1 を指定すると、経路の有効期限が生じます。これはデフォルト値です。このオプションでは、負の値は指定できません。

## routerevalidate

**目的:** これを指定すると、各接続のキャッシュに書き込まれた経路指定の妥当性を、新規経路指定が経路指定テーブルに追加されるたびに再検査する必要があります。これによって、長時間同じ接続をオープンしたままにするアプリケーション (NFS など) は、経路指定テーブルが変更された後に必ず正しい経路指定を使用できます。

**値:**

デフォルト: 0 (オフ)

範囲: 0 または 1

タイプ: Dynamic

**診断:** HACMP™ ノード間または NFS クライアントとサーバー間で通信上のミスが発生した場合、経路指定が変更されていた可能性があるが、キャッシュ内の値を使おうとします。

**チューニング:**

デフォルト値 0 の場合は、キャッシュに書き込まれた経路の妥当性は再検査されません。このオプションをオンにすると、ある程度パフォーマンスが低下します。HACMP の場合は、値を 1 に設定する必要があります。

## rto\_high

**目的:** ファクターの計算に使用される TCP 再送タイムアウトの上限値と、TCP データ・セグメントの再送時に使用される最大許容再送回数を指定します。**rto\_high** パラメーターは上限ファクターです。その値は偶数でなければなりません。**rto\_high** に奇数が設定されている場合、デフォルトとして値は 64 になります。デフォルト値は 64 です。**rto\_high** はロード時属性です。

**値:**

デフォルト: 64

範囲: 2 から MAXLONG

タイプ: Reboot

**診断:** 該当しない

**チューニング:**

**rto\_high** パラメーターは上限ファクターです。

## rto\_length

**目的:** ファクターの計算に使用される TCP 再送タイムアウトの長さと、TCP データ・セグメントの再送時に使用される最大許容再送回数を指定します。

**値:**

デフォルト: 13

範囲: 1 から 64

タイプ: Reboot

**診断:** 該当しない

**チューニング:**

**rto\_length** パラメーターは、時間セグメントの合計数です。

## rto\_limit

**目的:** ファクターの計算に使用される TCP 再送タイムアウトの限界値と、TCP データ・セグメントの再送時に使用される最大許容再送回数を指定します。

**値:**

デフォルト: 7

範囲: 1 から 64

タイプ: Reboot

**診断:** 該当しない

**チューニング:**

**rto\_limit** パラメーターは、**rto\_low** から **rto\_high** までの時間セグメントの数です。

## rto\_low

**目的:** ファクターの計算に使用される TCP 再送タイムアウトの下限値と、TCP データ・セグメントの再送時に使用される最大許容再送回数を指定します。**rto\_low** パラメーターは下限ファクターです。その値は 1 または偶数でなければなりません。**rto\_low** に奇数が設定されている場合、デフォルトとして値は 1 になります。デフォルト値は 1 です。**rto\_low** はロード時属性です。

**値:**

デフォルト: 1

範囲: 1 から 64

タイプ: Reboot

**診断:** 該当しない

**チューニング:**

**rto\_low** パラメーターは下限ファクターです。



## sack (AIX 4.3.3 以降)

**目的:** Selective Acknowledgment がオンかどうかを指定します。

**値:**

デフォルト: 0 (オフ)

範囲: 0 または 1

タイプ: Connect

**診断:** 該当しない

**チューニング:**

通常の TCP は、ドロップされたパケットの後はすべてのパケットを再送する必要があります。ウィンドウ・サイズが大きい場合、多くのパケットがドロップされると、パフォーマンスの悪化につながります。Selective Ack は、脱落したパケットをコードが選択して再送できるようにします。フラグメントの再送が多く、受信バッファのサイズが大きい場合は、このパラメーターをオンにすると効果がある場合があります。これは TCP の交渉オプションなので、使用するには両端でこれをサポートする必要があります。

## sb\_max

**目的:** TCP および UDP ソケットで可能な最大バッファ・サイズを指定します。setsockopt、udp\_sendspace、udp\_recvspace、tcp\_sendspace、および tcp\_recvspace を制限します。

**値:**

デフォルト: 1048576 バイト

範囲: 1 から MAXLONG

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

サイズを、できれば 4096 の倍数に増やします。最大ソケット・バッファ制限のおよそ 2 倍から 4 倍にすべきです。

**参照:** TCP streaming workload tuning のセクション。

## send\_file\_duration

**目的:** ネットワーク・バッファ・キャッシュ内でシステム・コール send\_file がアクセスしたすべてのファイル・オブジェクトに関して、キャッシュ妥当性検査の期間を指定します。

**値:**

デフォルト: 300 (5 分)

範囲: 0 から UINTMAX

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

0 は、アクセスごとにキャッシュが妥当性検査されることを意味します。このパラメーターは、AIX 4.3.2 以降にのみ適用されます。

## site6\_index

**目的:** サイト・ローカル経路指定に関して最大インターフェース番号を指定します。

**値:**

デフォルト: 0

範囲: 0 から MAXSHORT

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## sockthresh

**目的:** ソケットに割り当てることができるネットワーク・メモリーの最大量を指定します。新しいソケットまたは TCP 接続がすべての MBUF メモリーを使い果たすことを防ぎ、既存のソケットまたは TCP 接続用に残りのメモリーを予約するために使用します。

**値:**

デフォルト: 85%

範囲: 0 から 100

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

**net\_malloc** サブルーチンによって割り当てられたメモリーの合計量がこのしきい値に達すると、**socket** システム・コールと **socketpair** システム・コールは、エラー ENOBUFS が生じて失敗します。着信接続要求は、警告なしに廃棄されます。既存ソケットは、引き続き追加メモリーを使用できます。sockthresh パラメーターは、thewall パラメーターのパーセントを示し、可能な値は 1 から 100 で、デフォルトは 85 です。

**参照:** Tuning mbuf pool performance のセクション。

## sodebug

**目的:** 新たに作成されたソケットが、オンの SO\_DEBUG フラグを持つかどうかを指定します。

**値:**

デフォルト: 0 (no)

範囲: 0 または 1 (yes)

タイプ: Connect

**診断:** 該当しない

**チューニング:**

該当しない

## sodebug\_env

**目的:** 新たに作成されたソケットの SODEBUG プロセス環境変数を検査するかどうかを指定します。

**値:**

デフォルト: 0 (no)  
範囲: 0 または 1 (yes)  
タイプ: Connect

**診断:** 該当しない

**チューニング:**

該当しない

## somaxconn

**目的:** 最大 listen バックログを指定します。

**値:**

デフォルト: 1024 接続  
範囲: 0 から MAXSHORT  
タイプ: Connect

**診断:** 該当しない

**チューニング:**

ピーク時の接続率を処理するため、負荷の高い Web サーバーでこのパラメーターを増やします。

## subnetsarelocal

**目的:** サブネット・マスクに一致するすべてのサブネットを、確立の観点から (例えば、TCP 最大セグメント・サイズ) ローカルと見なすかどうかを指定します。

**値:**

デフォルト: 1 (yes)  
範囲: 0 または 1  
タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

このパラメーターは **in\_localaddress** サブルーチンで使用されます。デフォルト値 1 は、ローカル・ネットワーク・マスクと一致するアドレスがローカルであることを指定します。値 0 の場合は、ローカル・サブネットワークと一致するアドレスだけがローカルになります。これは、パフォーマンスに関する構成上の決定です。サブネットがすべて同じ MTU を持たない場合、ブリッジでのフラグメント化によりパフォーマンス低下が起きます。サブネットが同じ MTU を持ち、**subnetsarelocal** が 0 の場合、TCP セッションは不必要に小さい MSS を使用することになります。

**参照:** TCP Maximum Segment Size tuning のセクション。

## tcp\_bad\_port\_limit

**目的:** 500 ミリ秒の間に、ソケットのないポートに TCP パケットをいくつ受け取ったら、TCP はこのようなパケットに回答するリセットの送信を停止するかを示す、TCP パケット数を指定します。これをデフォルト値の 0 にすると、正しくないポート番号で TCP パケットを受信した場合、常にリセットが送信されます。

**値:**

デフォルト: 0

範囲: 0 から MAXLONG

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## tcp\_ecn

**目的:** RFC 2481 で記述されているように、明示的輻輳 (ふくそう) 通知の TCP レベル・サポートを可能にします。デフォルトはオフ (0) です。この値をオン (1) にすると、すべての接続はピアとの ECN 機能をネゴシエーションします。この機能を作動させるには、ピア TCP からのサポートとパス内のルーターからの IP レベル ECN サポートも必要です。

**値:**

デフォルト: 0 (オフ)

範囲: 0 または 1 (オン)

タイプ: Connect

**診断:** 該当しない

**チューニング:**

該当しない

## tcp\_ephemeral\_high

**目的:** TCP 一時ポートに割り当てる最大ポート番号を指定します。

**値:**

デフォルト: 65535

範囲: 32769 から 65535

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

一時ソケットの数は、**tcp\_ephemeral\_high** から **tcp\_ephemeral\_low** を引いた数です。一時ソケットの最大数については、**tcp\_ephemeral\_high** を 65535 に、**tcp\_ephemeral\_low** を 1024 にしてください。

## tcp\_ephemeral\_low

**目的:** TCP 一時ポートに割り当てる最小ポート番号を指定します。

**値:**

デフォルト: 32768

範囲: 1024 から 65534

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

一時ソケットの数は、**tcp\_ephemeral\_high** から **tcp\_ephemeral\_low** を引いた数です。一時ソケットの最大数については、**tcp\_ephemeral\_high** を 65535 に、**tcp\_ephemeral\_low** を 1024 にしてください。

## tcp\_finwait2

**目的:** 接続をクローズする前に FIN\_WAIT2 状態で待つ時間 (0.5 秒単位) を指定します。

**値:**

デフォルト: 1200 (600 秒または 10 分)

範囲: 0 から MAXSHORT

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## tcp\_icmpsecure

**目的:** TCP への ICMP (Internet Control Message Protocol) アタックを回避するかどうかを指定します。

**値:**

デフォルト: 0 (オフ)

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

TCP 接続を ICMP アタックから保護するために、このオプションはオンにすべきです。ICMP アタックの形式は、ICMP 発信元アタックおよび PMTUD (Path MTU Discovery) アタックとなります。このネットワーク・オプションがオンにされると、システムは ICMP 発信元メッセージに反応しません。これにより、ICMP 発信元アタックが回避されます。また、このネットワーク・オプションが使用可能になっていると、ICMP メッセージのペイロードがテストされ、ペイロードの TCP ヘッダー部分の順序番号が受け入れ可能な順序番号の範囲内にあるかどうか判別されます。これにより PMTUD アタックが大幅に緩和されます。

## tcp\_init\_window

**目的:** この値は、rfc2414 がオンにされた場合だけ使用されます (その他の場合は無視されます)。 rfc2414 がオンで、この値が 0 の場合は、初期ウィンドウの計算は rfc2414 に従って行われます。この値が 0 以外の場合、初期 (輻輳) ウィンドウは最大サイズのセグメントの数として **tcp\_init\_window** と等しい値に初期化されます。

**値:**

デフォルト: 0

範囲: 0 から MAXSHORT

タイプ: Connect

**診断:** 該当しない

**チューニング:**

**tcp\_init\_window** を変更すると、「TCP スロー・スタート」を調整して、ACK を受信する前の未解決の TCP セグメント (パケット) の数を制御することが可能となります。例えば、この値を 6 に設定すると、通常の 2 または 3 パケットではなくて、最初から 6 パケットの送信が可能となり、結果として、最初のパケット速度が早くなります。

## tcp\_inpcb\_hashtab\_siz

**目的:** TCP 接続の **inpcb** ハッシュ・テーブルのサイズを指定します。このテーブルは接続管理に必要な **inpcbs** を保持し、ハッシュ・チェーンのテーブルとしてインプリメントされます。大きなテーブルは、リンクされたハッシュ・チェーンが小さく、平均トラバース時間を下回るが、メモリー・フットプリントは大きくなることを意味します。

**値:**

デフォルト: 24499

範囲: 1 から 999999

タイプ: Reboot

**診断:** 該当しない

**チューニング:**

このオプションはパフォーマンスに影響を及ぼすため、特に注意して使用する必要があります。この値を変更する必要がある場合は、パフォーマンス分析者に相談してください。実行環境がこの値に影響を及ぼす可能性があります。システム定義のデフォルトを維持することを強くお勧めします。これは、このデフォルトがほとんどの環境で最適に実行される傾向があるためです。

## tcp\_keepcnt

**目的:** キープアライブ・プローブをいくつ送信したら接続を終了するかを示す数を指定します。

**値:**

デフォルト: 8

範囲: 0 から MAXSHORT

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

<b>tcp_keepidle</b>	<p><b>目的:</b> アイドル TCP 接続をアクティブにしておく時間 (0.5 秒単位) を指定します。</p> <p><b>値:</b></p> <p style="padding-left: 20px;">デフォルト: 14400 (2 時間)</p> <p style="padding-left: 20px;">範囲: 1 から MAXSHORT</p> <p style="padding-left: 20px;">タイプ: Connect</p> <p><b>診断:</b> 該当しない</p> <p><b>チューニング:</b> これは、わずかなパフォーマンス向上を求める構成上の決定です。変更しないことをお勧めします。</p>
<b>tcp_keepinit</b>	<p><b>目的:</b> TCP 接続の初期タイムアウト値を 0.5 秒単位で設定します。</p> <p><b>値:</b></p> <p style="padding-left: 20px;">デフォルト: 150 (75 秒)</p> <p style="padding-left: 20px;">範囲: 1 から MAXSHORT</p> <p style="padding-left: 20px;">タイプ: Dynamic</p> <p><b>診断:</b> 該当しない</p> <p><b>チューニング:</b> 該当しない</p>
<b>tcp_keepintvl</b>	<p><b>目的:</b> TCP 接続を検査するため、何秒おきにパケットを送信するかの間隔を、0.5 秒単位で指定します。</p> <p><b>値:</b></p> <p style="padding-left: 20px;">デフォルト: 150 (75 秒)</p> <p style="padding-left: 20px;">範囲: 1 から MAXSHORT</p> <p style="padding-left: 20px;">タイプ: Connect</p> <p><b>診断:</b> 該当しない</p> <p><b>チューニング:</b> これは、わずかなパフォーマンス向上を求める構成上の決定です。変更しないことをお勧めします。間隔を極端に短くすると、処理と帯域幅に多大な負担がかかります。</p>
<b>tcp_limited_transmit</b>	<p><b>目的:</b> RFC 3042 に記述して、TCP の欠落のリカバリーを拡張する機能を使用可能にします。</p> <p><b>値:</b></p> <p style="padding-left: 20px;">デフォルト: 1 (オン)</p> <p style="padding-left: 20px;">範囲: 0 または 1</p> <p style="padding-left: 20px;">タイプ: Dynamic</p> <p><b>診断:</b> 該当しない</p> <p><b>チューニング:</b> 該当しない</p>



## tcp\_low\_rto

**目的:** パケット・ドロップが発生しだしたすべての TCP 接続の RTO (単位: ミリ秒) を指定します。

**値:**

デフォルト: 0 (秒)

範囲: 0-3000 (ミリ秒)

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

値セットは **timer\_wheel\_tick** 値と等しいか、またはその 10 の倍数でなければなりません。 **tcp\_low\_rto** オプションの設定前は、現在値はゼロ以外の値でなければなりません。

## tcp\_maxburst

**目的:** バックツールバックのパケットを TCP がいくつ送信したら、このようなパケットを宛先に転送することを止めるかを示す数を指定します。これは、ルーターが大量の TCP パケットの洪水を処理できず、そのいくつかをドロップしているような場合に役に立ちます。値 0 は、バックツールバックのパケットについて制限がないことを意味します。

**値:**

デフォルト: 0

範囲: 0 から MAXSHORT

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## tcp\_mssdfit

**目的:** リモート・ネットワークとの通信に使用するデフォルトの最大セグメント・サイズ。

**値:**

デフォルト: 1460 バイト

範囲: 1 から USHORTMAX

タイプ: Connect

**診断:** 該当しない

**チューニング:**

AIX 4.2.1 以降では、**tcp\_mssdfit** が使用されるのは、パス MTU ディスカバリーが使用可能でない場合か、またはパス MTU ディスカバリーがパス MTU を見つけることができない場合だけです。データを (MTU - 40) バイトに制限することで、可能な場合、いっぱいであるパケットだけが送信されます。

**参照:** TCP Maximum Segment Size tuning のセクション。

## tcp\_nagle\_limit

**目的:** これはバイト単位の Nagle アルゴリズムしきい値で、Nagle を使用不可にするために使用できます。

**値:**

デフォルト: 65535 - IP パケットの最大サイズ

範囲: 0 から 65535

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

デフォルト (65535 - IP パケットの最大サイズ) は、オンにされた Nagle です。Nagle を使用不可にするには、この値を 0 か 1 にします。TCP はこのしきい値以上のデータ・セグメントの Nagle を使用不可にします。

## tcp\_nagleoverride

**目的:** 接続中の特定の状態に対して Nagle アルゴリズムを使用不可にします。Nagle アルゴリズムをオフにすることについて詳しくは、**tcp\_nagle\_limit** オプションを参照してください。特定の接続に対して Nagle アルゴリズムをオフにすることについて詳しくは、**ifconfig** コマンドの **tcp\_nodelay** オプションを参照してください。

**値:**

デフォルト: 0 (オフ)

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

1 の値を指定すると、接続で特定の TCP パケットに対する Nagle アルゴリズムが使用不可になります。

## tcp\_ndebug

**目的:** **tcp\_debug** 構造の数を指定します。

**値:**

デフォルト: 100

範囲: 0 から MAXSHORT

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## tcp\_newreno

**目的:** RFC 2582 に記述したように、TCP の Fast Recovery アルゴリズムへの修正を可能にします。これは TCP の Fast Retransmit アルゴリズムの制限を固定し、ウィンドウ内の複数のパケットが除去された場合、除去されたパケットから速くりカバーさせます。sack も同じ結果になりますが、sack には TCP 接続の両端からのサポートが必要です。NewReno 修正は送信側だけで行われます。

**値:**

デフォルト: 1 (オン)

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## tcp\_nodelayack

**目的:** このパラメーターをオンにすると、TCP は即時肯定応答 (Ack) パケットを送信側に送ります。**tcp\_nodelayack** がオフの場合は、TCP は Ack パケットの送信を最大 200 ミリ秒まで遅らせます。これにより、Ack を応答に結合することができ、システムのオーバーヘッドが最小化されます。

**値:**

デフォルト: 0 (オフ)

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

このオプションを使用して、TCP nagle アルゴリズムの他のインプリメンテーションでのバグを克服することができます。このオプションを 1 に設定すると、システムのオーバーヘッドが少し増えますが、送信側が受信側の肯定応答を待っている場合は、ネットワーク転送に関するパフォーマンスが著しく向上する可能性があります。

## tcp\_pmtu\_discover

**目的:** TCP アプリケーションのパス MTU ディスカバリーを使用可能または使用不可にします。

**値:**

デフォルト: 1 (AIX 4.3.3 より前は 0)

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

値が 0 の場合、TCP アプリケーションのパス MTU ディスカバリーは使用不可になり、値が 1 の場合は使用可能になります。このパラメーターは、AIX 4.2.1 以降にのみ適用されます。

**参照:** TCP Maximum Segment Size tuning のセクション。

## tcp\_recvspace

**目的:** データを受信するためのシステムのデフォルト・ソケット・バッファ・サイズを指定します。これは TCP によって使用されるウィンドウ・サイズに影響します。

**値:**

デフォルト: 16384 バイト

範囲: 4096 から MAXLONG

タイプ: Connect

**診断:** 該当しない

**チューニング:**

ソケット・バッファ・サイズを 16 キロバイト (16,384) に設定すると、標準イーサネットおよびトークンリング・ネットワークにおけるパフォーマンスが向上します。シリアル回線インターネット・プロトコル (SLIP) などの帯域幅の狭いネットワーク、または、シリアル・オプティカル・リンクなどの帯域幅の広いネットワークでは、最適バッファ・サイズは異なります。最適なバッファ・サイズはメディアの帯域幅とパケットの平均往復時間の積です。

高速ネットワーク (例えば、ギガビット・イーサネットや ATM 622 など) の場合は、最良のパフォーマンスを確保するための最小サイズに値 65536 を使用する必要があります。

**tcp\_recvspace** パラメーターには、**sb\_max** パラメーターの設定値以下のソケット・バッファ・サイズを指定する必要があります。インターフェースごとに **ISNO** オプションを使用すると、このグローバル設定値もオーバーライドされます。65536 より大きい値の場合は、TCP ウィンドウ・スケーリングを使用可能にするために **rfc1323 (rfc1323=1)** を使用可能にする必要があります。

**参照:** TCP workload tuning のセクション。

## tcp\_sendspace

**目的:** データを送信するためのシステムのデフォルト・ソケット・バッファ・サイズを指定します。

**値:**

デフォルト: 16384 バイト

範囲: 4096 から MAXLONG

タイプ: Connect

**診断:** 該当しない

**チューニング:**

これは TCP によって使用されるウィンドウ・サイズに影響します。ソケット・バッファ・サイズを 16 キロバイト (16,384) に設定すると、標準イーサネットおよびトークンリング・ネットワークにおけるパフォーマンスが向上します。

シリアル回線インターネット・プロトコル (SLIP) などの帯域幅の狭いネットワーク、または、シリアル・オプティカル・リンクなどの帯域幅の広いネットワークでは、最適バッファ・サイズは異なります。最適なバッファ・サイズはメディアの帯域幅とパケットの平均往復時間の積です。

$optimum\_window = bandwidth * average\_round\_trip\_time$

高速ネットワーク (例えば、ギガビット・イーサネットや ATM 622 など) の場合は、最良のパフォーマンスを確保するための最小サイズに値 65536 を使用する必要があります。

**tcp\_sendspace** パラメーターには、**sb\_max** パラメーターの設定値以下のソケット・バッファ・サイズを指定する必要があります。インターフェースごとに **ISNO** オプションを使用すると、このグローバル設定値もオーバーライドされます。65536 より大きい値の場合は、TCP ウィンドウ・スケーリングを使用可能にするために **rfc1323** (**rfc1323=1**) を使用可能にする必要があります。

**参照:** TCP workload tuning のセクション。

## tcp\_tcpsecure

**目的:** TCP における接続リセット・アタックおよびデータ破壊アタックを回避するかどうかを指定します。

**値:**

デフォルト: 0 (オフ)

範囲: 0 から 7

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

このオプションは、以下の 3 つのぜい弱性の 1 つ以上から TCP 接続を保護する場合に使用します。最初のぜい弱性には、接続を異常終了させるための、確立済みの接続に対する偽の SYN の送信が含まれます。

**tcp\_tcpsecure** 値が 1 の場合は、このぜい弱性から保護します。2 番目のぜい弱性には、接続を異常終了させるための、確立済みの接続に対する偽の RST の送信が含まれます。**tcp\_tcpsecure** 値が 2 の場合は、このぜい弱性から保護します。3 番目のぜい弱性には、確立済みの TCP 接続における偽データの注入が含まれます。**tcp\_tcpsecure** 値が 4 の場合は、このぜい弱性から保護します。**tcp\_tcpsecure** の値の範囲は、最小の 0 (これがデフォルト値で、この場合はこれらのぜい弱性からの保護はありません) から最大の 7 までです。3、5、6、または 7 の値は、接続をこれら 3 つのぜい弱性の組み合わせから保護します。

## tcp\_timewait

**目的:** **tcp\_timewait** オプションは、接続を timewait 状態に保つ時間を構成し、15 秒間隔で指定します。

**値:**

デフォルト: 1

範囲: 1 から 5

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

この値を増やすと、多数の TCP 接続をオープン/クローズする Web サーバーやアプリケーションのパフォーマンスが悪化します。

## tcp\_ttl

**目的:** TCP パケットの存続時間を指定します。

**値:**

デフォルト: 60 ホップ

範囲: 1 から 255

タイプ: Connect

**診断:** この値は、TCP パケットの IP ヘッダーでの存続時間 (**ip\_ttl**) となります。

**チューニング:**

該当しない

## tcprexttthresh

**目的:** TCP を高速再送フェーズに入らせる、連続する重複肯定応答の数を指定します。

**値:**

デフォルト: 3

範囲: 1 から MAXSHORT

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

重複肯定応答の数が増えたことにより TCP のパフォーマンスが低い場合 (ただし、ネットワークは混み合っていない)、このパラメーターを大きくします。このオプションに対して高い値を設定すると、TCP がタイムアウトし、再送の原因となることがあることに注意してください。

## thewall

**目的:** メモリー・プールに割り当てられる最大メモリー量をキロバイト単位で指定します。

**値:**

デフォルト: AIX バージョン 5.1 以降:

64 ビット・カーネルの場合は RAM の 1/2

32 ビット・カーネルの場合、RAM の 1/2 または 1 GB (どちらか小さい方) です。

範囲: 該当しない

タイプ: Static

**診断:** 該当しない

**チューニング:**

AIX 5.1 以降では設定できません。

**参照:** Network performance analysis のセクション。

## timer\_wheel\_tick

**目的:** タイマー・ホイールのスロット間隔をティック (刻み)で指定します。ここで 1 ティックは 10 ミリ秒です。このオプションを変更した場合は、変更した値を有効にするにはシステムをリブートしてください。

**値:**

デフォルト: 0

範囲: 0-100

タイプ: Reboot

**診断:** 該当しない

**チューニング:**

実際のスロット・インターバルを得るには、指定した値に 10 ミリ秒を乗算しなければなりません。



## udp\_bad\_port\_limit

**目的:** 500 ミリ秒の間に、ソケットのないポートに UDP パケットをいくつ受け取ったら、UDP はこのようなパケットに回答する ICMP エラーの送信を停止するかを示す、UDP パケット数を指定します。これを 0 にすると、正しくないポート番号で UDP パケットを受信した場合、常に ICMP エラーが送信されます。0 より大きい値は、この数のパケットを受信したら UDP は ICMP エラーの送信を停止することを示す数です。

**値:**

デフォルト: 0

範囲: 0 から MAXLONG

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## udp\_ephemeral\_high

**目的:** UDP 一時ポートに割り当てる最大ポート番号を指定します。

**値:**

デフォルト: 65535

範囲: 32769 から 65535

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## udp\_ephemeral\_low

**目的:** UDP 一時ポートに割り当てる最小ポート番号を指定します。

**値:**

デフォルト: 32768

範囲: 1 から 65534

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## udp\_inpcb\_hashtab\_siz

**目的:** UDP 接続の **inpcb** ハッシュ・テーブルのサイズを指定します。このテーブルは接続管理に必要な **inpcbs** を保持し、ハッシュ・チェーンのテーブルとしてインプリメントされます。大きなテーブルは、リンクされたハッシュ・チェーンが小さく、平均トラバース時間を下回るが、メモリー・フットプリントは大きくなることを意味します。

**値:**

デフォルト: 24499

範囲: 1 から 83000

タイプ: Reboot

**診断:** 該当しない

**チューニング:**

このオプションはパフォーマンスに影響を及ぼすため、特に注意して使用する必要があります。この値を変更する必要がある場合は、パフォーマンス分析者に相談してください。実行環境がこの値に影響を及ぼす可能性があります。システム定義のデフォルトを維持することを強くお勧めします。これは、このデフォルトがほとんどの環境で最適に実行される傾向があるためです。

## udp\_pmtu\_discover

**目的:** UDP アプリケーションのパス MTU ディスカバリーを使用可能または使用不可にします。

**値:**

デフォルト: 1 (AIX 4.3.3 より前は 0)

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

UDP アプリケーションは、パス MTU ディスカバリーを使用することを特定して書く必要があります。値が 0 の場合、この機能は使用不可になり、値が 1 の場合は使用可能になります。デフォルト値は 0 です。このパラメーターは AIX バージョン 4.2.1 以降にのみ適用されます。

## udp\_recvspace

**目的:** UDP データを受信するためのシステムのデフォルト・ソケット・バッファ・サイズを指定します。

**値:**

デフォルト: 42080 バイト

範囲: 4096 から MAXLONG

タイプ: Connect

**診断:** **netstat -s** のゼロでない  $n$  は、**udp** の報告です。 $n$  ソケット・バッファがオーバーフローしています。

**チューニング:**

**udp\_recvspace** パラメーターには、**sb\_max** パラメーターの設定値以下のソケット・バッファ・サイズを指定する必要があります。サイズを、できれば 4096 の倍数に増やします。

**参照:** UDP tuning のセクション。

## udp\_sendspace

**目的:** UDP データを送信するためのシステムのデフォルト・ソケット・バッファ  
ー・サイズを指定します。

**値:**

デフォルト: 9216 バイト

範囲: 4096 から MAXLONG

タイプ: Connect

**診断:** 該当しない

**チューニング:**

**udp\_sendspace** パラメーターには、**sb\_max** パラメーターの設定値以  
下のソケット・バッファ・サイズを指定する必要があります。サイズ  
を、できれば 4096 の倍数に増やします。

**参照:** UDP tuning のセクション。

## udp\_ttl

**目的:** UDP パケットの存続時間を指定します。

**値:**

デフォルト: 30 秒

範囲: 1 から 255

タイプ: Connect

**診断:** この値は、UDP パケットの IP ヘッダーでの存続時間 (**ip\_ttl**) となりま  
す。

**チューニング:**

該当しない

## udpcksum

**目的:** UDP チェックサムのオン/オフを切り替えることができます。

**値:**

デフォルト: 1

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

値が 0 の場合はオフになり、値が 1 の場合はオンになります。

## use\_isno

**目的:** インターフェースごとのチューニング・オプションを可能にします。

**値:**

デフォルト: 1 (オン)

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

値 1 はオンであり、0 はオフです。インターフェースごとの TCP チューナブル・パラメーター (SMIT または `chdev` コマンドによりチューニング可能) が設定されている場合、`use_isno` を 1 にすると TCP グローバル値をオーバーライドします。その場合でも、アプリケーションは **setsockopt** サブルーチンにより、これらすべてをオーバーライドすることができます。

**参照:** インターフェース固有のネットワーク・オプション (ISNO)

## use\_sndbufpool

**目的:** パフォーマンスを改善するために、MBUF クラスターのキャッシングを使用可能にします。この値が使用不可の場合、MBUF クラスターを割り当てるには、AIX はクラスター・バッファおよびそのクラスター・バッファを指示する MBUF バッファを割り当てる必要があり、つまり 2 つのバッファ割り当て操作が必要になります。同じく、クラスターを解放するには、2 つのバッファ解放操作が必要です。このオプションを使用可能にすれば、AIX は、クラスターのキャッシュを、使用中の各クラスター・サイズごとに保守します。こうして MBUF クラスターの割り当てと解放のためのオーバーヘッドを削減することにより、パフォーマンスを改善します。

**値:**

デフォルト: 1 (オン)

範囲: 0 または 1

タイプ: Reboot

**診断:** 該当しない

**チューニング:**

デフォルト値 1 は、このオプションをシステム全体規模で使用可能にします。MBUF クラスター・キャッシュは、**netstat -M** コマンドを使用して表示することができます。

**参照:** `Tuning mbuf pool performance` のセクション。

## ストリーム・チューナブル・パラメーター

### lowthresh

**目的:** BPRI\_LO 優先順位の **allocb** コールを使用して割り当てることができる最大バイト数を指定します。

**値:**

デフォルト: 90 (thewall のパーセント)

範囲: 0 から 100

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

**net\_malloc** コールによって割り当てられたメモリーの合計量がこのしきい値に達すると、BPRI\_LO 優先順位の **allocb** 要求は 0 を戻します。**lowthresh** パラメーターは、thewall パラメーターのパーセントを表し、0 から 100 までの値を設定できます。

### medthresh

**目的:** BPRI\_MED 優先順位の **allocb()** コールを使用して割り当てることができる最大バイト数を指定します。

**値:**

デフォルト: 95 (thewall のパーセント)

範囲: 0 から 100

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

**net\_malloc** コールによって割り当てられたメモリーの合計量がこのしきい値に達すると、BPRI\_MED 優先順位の **allocb** 要求は 0 を戻します。**medthresh** パラメーターは、thewall パラメーターのパーセントを表し、0 から 100 までの値を設定できます。

### nstrpush

**目的:** 単一のストリームに送信できるモジュールの最大数 (最低 8) を指定します。

**値:**

デフォルト: 8

範囲: 8 から MAXSHORT

タイプ: Static

**診断:** 該当しない

**チューニング:**

AIX 5.2 以降では読み取り専用。この属性は、ブート時に **/etc/pse\_tune.conf** ファイルに設定することができます。

## psebufcalls

**目的:** ストリームによって割り当てられる `bufcall` の最大数を指定します。

**値:**

デフォルト: 20

範囲: 20 から MAXLONG

タイプ: Incremental

**診断:** 該当しない

**チューニング:**

ストリーム・サブシステムは、初期化時に一定数の `bufcall` 構造を割り当てます。`allocb` コールが失敗したときに、ユーザーは `bufcall` への要求を登録できます。システムがリブートしてデフォルト値に戻すまで、この値を下げることはできません。

## psecache

**目的:** ストリーム・バッファの数を制御します。

**値:**

デフォルト: 1

範囲: 0 または 1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

## pseintrstack

**目的:** `offlevel` で実行中にストリームによって実行できる割り込みスタックの最大サイズを指定します。

**値:**

デフォルト: 32 ビット・マシンでは 0x3000、64 ビット・マシンでは 0x6000。

範囲: 12288 から MAXLONG

タイプ: Static

**診断:** 該当しない

**チューニング:**

AIX 5.2 以降では読み取り専用。この属性は、ブート時に `/etc/pse_tune.conf` ファイルに設定することができます。INTBASE レベル以外で稼働しているプロセスがストリームに入った場合、割り込みスタック・サイズが小さすぎるとスタック・オーバーフローの問題が起こることがあります。このパラメーターを適切に設定することによって、スタック・オーバーフローの発生を減らすことができます。

## psetimers

**目的:** ストリームによって割り当てられるタイマーの最大数を指定します。

**値:**

デフォルト: 20

範囲: 20 から MAXLONG

タイプ: Incremental

**診断:** 該当しない

**チューニング:**

ストリーム・サブシステムは初期設定時に一定数のタイマー構造を割り当てるので、ストリーム・ドライバーまたはモジュールは **timeout** コールを登録できます。システムがリポートしてデフォルト値に戻すまで、この値を下げることはできません。

## strctlsz

**目的:** 単一のシステム・コールでストリームに渡してメッセージのコントロール部分 (M\_PROTO または M\_PCPROTO ブロック) に置くことができる情報の最大バイト数を指定します。

**値:**

デフォルト: 1024

範囲: 0 から MAXSHORT

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

**putmsg** コールでコントロール部分がこのサイズを超えると、ERANGE で失敗します。

## strmsgsz

**目的:** 単一のシステム・コールでストリームに渡してメッセージのデータ部分 (M\_DATA ブロック) に置くことができる情報の最大バイト数を指定します。

**値:**

デフォルト: 0

範囲: 0 から MAXSHORT

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

**write** コールでこのサイズを超えると、複数のメッセージに分割されます。 **putmsg** コールでデータ部分がこのサイズを超えると、ERANGE で失敗します。



## strthresh

**目的:** ストリームが通常割り当てることができる最大バイト数を、`thewall` のパーセントで指定します。

**値:**

デフォルト: **thewall** の 85%

範囲: 0 から 100

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

このしきい値に達した場合、必要な特権のないユーザーは、ストリームのオープン、モジュールのプッシュまたはストリーム・デバイスへの書き込みを実行できなくなり、ENOSR が戻されます。しきい値は出力側にのみ適用され、システムに入ってくるデータには影響しません (例えば、コンソールは正常に働きます)。値 0 は、しきい値がないことを意味します。 **strthresh** パラメーターは、**thewall** パラメーターのパーセントを表し、0 から 100 までの値を設定できます。 **thewall** パラメーターは、**net\_malloc** コールを使用してストリームおよびソケットによって割り当てられる最大バイト数を示します。

**参照:** Tuning mbuf pool performance のセクション。

## strturncnt

**目的:** Module または Elsewhere レベルのストリーム同期化用に、現在実行中のスレッドが扱える要求の最大数を指定します。

**値:**

デフォルト: 15

範囲: 1 から MAXLONG

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

Module レベルの同期では、モジュール内で実行できるスレッドは常に 1 つだけであり、他のすべてのスレッドは、同じモジュールを獲得しようとする、要求をエンキューして終了します。現在実行中のスレッドが作業を完了すると、その前にエンキューされた要求が 1 つずつデキューされ、呼び出されます。多数の要求がリストにエンキューされると、現在実行中のスレッドはそのすべてに対応しなければならず、その対応に追われ、動けなくなります。これを避けるため、現在実行中のスレッドは `strturncnt` で指定した数のスレッドだけに対応し、そのあとで別のカーネル・スレッドを起動し、すべての保留要求を呼び出します。

## 例

- mbuf プールの最大サイズを表示するには、次のように入力します。  
`no -o thewall`
- UDP パケットの存続時間をデフォルト・サイズにリセットするには、次のように入力します。  
`no -d udp_ttl`
- システムのデフォルトのソケット・バッファ・サイズを変更するには、次のように入力します。

- ```
no -r -o tcp_sendspace=32768
no -r -o udp_recvspace=32768
```
4. マシンを TCP/IP ネットワークでのインターネット作業ルーターとして使用するには、次のように入力します。
 

```
no -o ipforwarding=1
```
  5. **no** コマンドが管理するすべてのチューナブル・パラメーターの、現行値とリポート値、範囲、単位、タイプと依存関係をリストするには、次のように入力します。
 

```
no -L
```
  6. `udp_ephemeral_high` のヘルプ情報を表示するには、次のように入力します。
 

```
no -h udp_ephemeral_high
```
  7. `ip6srcrouteforward` を永久にオフにするには、次のように入力します。
 

```
no -p -o ip6srcrouteforward=0
```
  8. すべてのネットワーク・チューニング・パラメーターのリポート値をリストするには、次のように入力します。
 

```
no -r -a
```
  9. **no** コマンドが管理するすべてのチューナブル・パラメーターの、現行値とリポート値、範囲、単位、タイプと依存関係をリスト (スプレッドシート形式) するには、次のように入力します。
 

```
no -x
```
  10. サイズが 256 または 4096 である、タイプ `mbuf` または `socket` のすべての割り当ておよび解放をログするには、次のように入力します。
 

```
no -o net_buf_type={mbuf:socket} -o net_buf_size={256:4096} -o net_malloc_police=1
```
  11. タイプ `mbuf` のすべての割り当ておよび解放をログするには、次のように入力します。
 

```
no -o net_buf_type={mbuf} -o net_buf_size={all} -o net_malloc_police=1
```
  12. 2000 イベント・バッファ・サイズを使用する `en0` または `en3` の、すべての **ns\_alloc** および **ns\_free** をログするには、次のように入力します。
 

```
no -o ndd_event_name={en0:en3} -o ndd_event_tracing=2000
```
  13. 2000 イベント・バッファ・サイズを使用する、すべての `en` アダプターのすべての **ns\_alloc** および **ns\_free** をログするには、次のように入力します。
 

```
no -o ndd_event_name={en} -o ndd_event_tracing=2000
```
  14. すべてのアダプターの **ns\_alloc** および **ns\_free** をログするには、次のように入力します。
 

```
no -o ndd_event_name={all} -o ndd_event_tracing=1
```

## 関連情報

**ifconfig** コマンド、**inetd** コマンド、**vmo** コマンド、**ioo** コマンド、**raso** コマンド、**schedo** コマンド、**tunsave** コマンド、**tunchange** コマンド、**tunrestore** コマンド、**tuncheck** コマンド、**tundefault** コマンド。

**setsockopt** サブルーチン。

通信およびネットワークのセクション。

TCP/IP のアドレッシング、TCP/IP のプロトコル、および TCP/IP の経路指定、インターネット・プロトコル、伝送制御プロトコル、インターネットのアプリケーション・レベル・プロトコル、および ユーザー・データグラム・プロトコルのセクション。

ネットワーク・パフォーマンスのセクション。

パス MTU ディスカバリーのセクション。

AIX 5L Version 5.3 Performance Tools Guide and Referenceのカーネル・チューニングのセクション。

AIX 5.2 互換モード。

---

## nohup コマンド

### 目的

コマンドをハングアップせずに実行します。

### 構文

```
nohup { -p pid | Command [ Arg ... ] [ & ] }
```

### 説明

**nohup** コマンドは、*Command* パラメーターおよび関連する *Arg* パラメーターで指定されたコマンドを、すべてのハングアップ (SIGHUP) シグナルを無視して実行するか、またはすべてのハングアップ (SIGHUP) シグナルを無視する **-p** オプションを使用して指定されたプロセスを変更します。

**nohup** コマンドは、ログオフ後にバックグラウンドでプログラムを実行するために使用することもできます。**nohup** コマンドをバックグラウンドで実行するには、**&** (アンパーサンド) をコマンドの終わりに追加します。

注: **-p pid** と *Command* を一緒に指定することはできません。

**-p pid** が使用される場合、指定されたプロセスの出力は **nohup.out** にはリダイレクトされません。

### フラグ

**-p pid** *pid* は、実行中のプロセスのプロセス ID です。**nohup** コマンドは指定されたプロセスを変更し、すべてのハングアップ (SIGHUP) シグナルを無視するようにします。

### 終了状況

このコマンドは、以下の終了値を戻します。

**126** *Command* パラメーターで指定されたコマンドを検出しましたが、呼び出すことができませんでした。  
**127** **nohup** コマンドまたは *Command* パラメーターで指定されたコマンドで発生したエラーを検出できませんでした。

それ以外の場合には、**nohup** コマンドの終了状況は、*Command* パラメーターで指定されたコマンドの終了状況になります。

### 例

1. ログアウト後にコマンドをバックグラウンドで実行するには、次のように入力します。

```
$ nohup find / -print &
```

このコマンドを入力すると、次のように表示されます。

```
670
$ Sending output to nohup.out
```

プロセス ID 番号は、& (アンパーサンド) によって始動されるバックグラウンド・プロセスのプロセス ID 番号に変わります。メッセージ「Sending output to nohup.out」は、**find / -print** コマンドからの出力が **nohup.out** ファイルに入っていることを示します。これらのメッセージが表示されたあとは、まだ **find** コマンドを実行中でもログアウトすることができます。

2. コマンドをバックグラウンドで実行し、標準出力を異なるファイルにリダイレクトするには、次のように入力します。

```
$ nohup find / -print >filenames &
```

この例では、**find / -print** コマンドが実行され、その出力がファイル `filenames` に格納されます。プロセス ID とプロンプトのみが次のように表示されます。

```
677
$
```

**nohup** コマンドが *Command* パラメーターで指定されたコマンドを始動するには時間がかかるので、ログアウトする前に少し待ってください。早くログアウトしすぎると、*Command* パラメーターで指定したコマンドが実行されないことがあります。*Command* パラメーターで指定したコマンドが始動されたら、ログアウトしても影響はありません。

3. 複数のコマンドを実行するにはシェル・プロシージャを使います。例えば、次のようにシェル・プロシージャを書いて

```
neqn math1 | nroff > fmath1
```

`nnfmath1` ファイルと命名します。この場合、次のコマンドを使うと、`nnfmath1` ファイル内のすべてのコマンドに **nohup** コマンドを実行できます。

```
nohup sh nnfmath1
```

4. 実行許可を `nnfmath1` ファイルに割り当てる場合、次のコマンドを発行すると、同じ結果を得ることができます。

```
nohup nnfmath1
```

5. `nnfmath1` ファイルをバックグラウンドで実行するには、次のように入力します。

```
nohup nnfmath1
&
```

6. `nnfmath1` ファイルを Korn シェルで実行するには、次のように入力します。

```
nohup ksh nnfmath1
```

7. 実行中のプロセスにすべてのハングアップ・シグナルを無視させるには、次のように入力します。

```
nohup -p 161792
```

## 関連情報

**cs** コマンド、**nice** コマンド、**sh** コマンド。

**signal** サブルーチン。

---

## notifyevent コマンド

### 目的

イベント応答リソース・マネージャー (ERRM) によって生成されたイベント情報を、指定されたユーザー ID にメールします。

### 構文

```
notifyevent [-h] [user-ID]
```

### 説明

**notifyevent** スクリプトは、イベントが発生したときにイベント応答リソース・マネージャー (ERRM) が生成する環境変数で、ERRM によって通知されるイベント情報をキャプチャーします。このスクリプトは、イベント応答リソースによって実行されるアクションとして使用することができます。また、他のユーザー定義アクションを作成するためのテンプレートとして使用することもできます。**notifyevent** スクリプトのメッセージが戻される言語は、ロケール設定により異なります。

ERRM 環境変数に関して戻されるイベント情報には、以下が含まれます。

#### Local Time

イベントまたはリアム・イベントが観察された時刻。ERRM が提供する実際の環境変数は `ERRM_TIME` です。この値はローカライズされ、読みやすい形式に変換されてから表示されます。

このスクリプトは **mail** コマンドを使用して、指定されたユーザー ID にイベント情報を送信します。ユーザー ID を指定すると、そのユーザー ID は有効であると見なされ、検証せずに使用されます。ユーザー ID を指定しないと、このコマンドを実行しているユーザーがデフォルトとして使用されます。

*user-ID* は、イベント情報のメール先ユーザーの ID です (オプション)。*user-ID* を指定しないと、このコマンドを実行しているユーザーがデフォルトとして使用されます。

### フラグ

**-h** スクリプトの使用ステートメント (使用法) を標準出力に書き込みます。

### パラメーター

#### *log\_file*

イベント情報のログを記録するファイルの名前を指定します。*log\_file* パラメーターの絶対パスを指定する必要があります。

*log\_file* は循環ログとして扱われ、64KB の固定サイズです。*log\_file* がいっぱいになると、既存の最も古いエントリー上に新しいエントリーが上書きされます。

*log\_file* が既に存在する場合、イベント情報はこれに付加されます。*log\_file* が存在しない場合は作成され、イベント情報を書き込めるようにします。

### 終了状況

0 コマンドが正常に実行されました

## 制約事項

1. これらのスクリプトは **ERRM** が実行されているノードで実行する必要があります。
2. **mail** コマンドを使用してファイルが読み取られます。

## 標準出力

**-h** フラグが指定されると、スクリプトの使用法ステートメントが標準出力に書き込まれます。

## 例

1. ユーザーにメールを送信するために、**Web-based System Manager** に **user1** を指定します。指定すると、イベント応答リソース・マネージャーは以下のコマンドを実行します。

```
/usr/sbin/rsct/bin/notifyscript user1
```

2. イベント情報の内容を読むには、**mail** コマンドを使用します。次の例は、**/var** ファイルシステム (ファイルシステム・リソース) の警告イベントがどのような形式で記録されるのかを示しています。

```
=====  
Event reported at Sun Mar 26 16:38:03 2007  
  
Condition Name:    /var space used  
Severity:         Warning  
Event Type:       Event  
Expression:       PercentTotUsed>90  
  
Resource Name:    /var  
Resource Class Name:  IBM.FileSystem  
Data Type:       CT_UINT32  
Data Value:      91
```

## 位置

**/usr/sbin/rsct/bin/notifyscript**

## 関連情報

コマンド: **mail**、**enotifyscript**

---

## nrglbd デーモン

### 目的

グローバル・ロケーション・ブローカー・データベースを管理します。

### 構文

**nrglbd** [ **-version** ]

### 説明

**glbd** デーモンは、グローバル・ロケーション・ブローカー (GLB) データベースを管理します。ネットワーク・コンピューティング・システム (NCS) の一部である GLB データベースは、クライアントがネットワークまたはインターネット上でサーバーを見つけるのを手伝います。GLB データベースは、処理が実行されているサーバーの位置 (つまりネットワーク・アドレスとポート番号) を格納します。**glbd** デーモンはこのデータベースを維持し、そこへアクセスできるようにします。

GLB デーモンには、**glbd** と **nrglbd** の 2 つのバージョンがあります。ネットワークまたはインターネットで実行できる **nrglbd** は 1 つだけです。また、**nrglbd** と **glbd** を同じネットワークまたはインターネット上で実行してはなりません。

通常、**nrglbd** デーモンはバックグラウンドで開始します。開始方法には、次の 2 つがあります。

- root ユーザー権限がある場合は、コマンド行に次のように入力します。

```
/etc/ncs/nrglbd &
```

- システム・リソース・コントローラー (SRC) を使って、コマンド行に次のように入力します。

```
startsrc -s nrglbd
```

**nrglbd** デーモンを開始する前に、TCP/IP を構成してシステムで作動させる必要があります。また、**nrglbd** デーモンを開始する前に、**llbd** デーモンも開始し、実行する必要があります。

## フラグ

**-version** この **nrglbd** が属す NCS のバージョンを表示します。ただし、デーモンの開始はしません。

## ファイル

**/etc/rc.ncs** NCS デーモンを開始するコマンドを含みます。

## 関連情報

**lb\_admin** コマンド。

**llbd** デーモン。

*AIX 5L Version 5.3 Communications Programming Concepts* の Location Broker のセクション。

---

## nroff コマンド

### 目的

タイプライターなどのデバイスやライン・プリンターへ出力するためにテキストをフォーマットします。

### 構文

```
nroff [ -e ] [ -h ] [ -i ] [ -q ] [ -z ] [ -o List ] [ -n Number ] [ -s Number ] [ -r ANumber ] [ -u Number ] [ -T Name ] [ -man ] [ -me ] [ -mm ] [ -mptx ] [ -ms ] [ File ... | - ]
```

### 説明

**nroff** コマンドは 1 つ以上のファイルを読み取り、タイプライターなどのデバイスおよびライン・プリンターに出力します。最後のパラメーターにファイルを指定しないか、**-** (マイナス記号) フラグを指定した場合、デフォルトで標準入力を読み取ります。*File* 変数には、**nroff** コマンドでタイプライターなどのデバイスに出力するファイルを指定します。デフォルトは標準入力です。

**nroff** コマンド出力をポストプロセスするには **col** コマンドが必要な場合があります。



## フラグ

- e** 特定の端末装置の全解像度を使って、行末調整済みの行に均等間隔のワードを生成します。
- h** 水平スペーシング時に出力タブ文字を使用して出力を迅速化し、出力文字数を少なくします。タブ文字は公称の 8 桁ごとに設定されるものと見なします。
- i** 指定ファイルをすべて読み取った後、標準入力を読み取ります。
- man** **man** マクロ処理パッケージを選択します。
- me** **me** マクロ処理パッケージを選択します。
- mm** **mm** マクロ処理パッケージを選択します。
- mptx** **mptx** マクロ処理パッケージを選択します。
- ms** **ms** マクロ処理パッケージを選択します。
- n Number** 指定された番号を最初の出力ページに割り当てます。
- o List** 下記のようなコンマで区切られたページ番号と範囲のリストで構成する、*List* 変数で指定されたページのみを出力します。
- 範囲 *Start-Stop* は、ページ *Start* から *Stop* までの出力を意味します。例えば、9-15 であれば 9 ページから 15 ページまでを出力します。
  - 最初の *-Stop* は、先頭ページからページ *Stop* までの出力を意味します。
  - 最後の *Start-* は、ページ *Start* から最後までを意味します。
  - ページ番号と範囲を組み合わせると、指定したページが出力されます。例えば、-3、6-8、10、12- であれば、先頭ページから 3 ページまで、6 ページから 8 ページまで、10 ページ、および 12 ページから最後までが出力されます。  
注: **-oList** フラグをパイプラインで使用する場合 (1 つ以上の **eqn** または **tbl** コマンドなど)、文書の最後のページを *List* パラメーターに指定しないと、「broken pipe」メッセージが表示されます。この「broken pipe」というメッセージは何も示しておらず、無視してもかまいません。
- q** **.rd** 要求の同時入出力モードを呼び出します。
- r ANumber** レジスター *A* を指定値に設定します。*A* 変数で指定した値は 1 文字の ASCII 名でなければなりません。
- s Number** 指定ページ数ごとに停止します (デフォルト値は 1 です)。**nroff** コマンドは、指定ページ数ごとに停止し、用紙を挿入したり、変更できるようにします。行送りまたは改行文字を受け取ると実行を再開します。このフラグはパイプラインでは働きません (例えば、**mm** コマンド)。**nroff** コマンドがページ間で停止するとき、ASCII BEL 文字がワークステーションに送られます。

**-TName**

指定したプリンター用の出力を作成します。タイプライターなどデバイスとライン・プリンターは、英語文字セット、数字、および記号の他に国際拡張文字セットを表す下記の *Name* 変数を使用します。

**hplj** Hewlett-Packard LaserJet II と同じシリーズの他のモデルのプリンター。

**ibm3812**

3812 ページ・プリンター II。

**ibm3816**

3816 ページ・プリンター。

**ibm4019**

4019 レーザー・プリンター。

**注:** 4019 および HP Laser Jet II プリンターはページの上部と下部に非出力域があります。このようなプリンターにファイルを出力する場合は、すべての出力を出力可能域に配置するように、上部マージンと下部マージンを定義しなければなりません (例えば、**-mm** フラグでフォーマットした場合)。

**37** 端末表示専用のテレタイプ・モデル 37 端末 (デフォルト)。このデバイスでは、**\[N]** フォーマットで入力された拡張文字がサポートされません。拡張単一バイト文字を入力すると、詳細情報が表示されます。

**lp** アンダーラインとタブを使用できるプリンターの総称名。逆方向の行送りを使用して **lp** 値に送信されるすべてのテキスト (例えば、テーブルを含むテキスト) は、**col** コマンドを使用して処理しなければなりません。このデバイスでは、**\[N]** フォーマットで入力された拡張文字がサポートされません。拡張単一バイト文字を入力すると、詳細情報が表示されません。

**ppds** Quietwriter III、Quickwriter、Proprinters など、パーソナル・プリンターのデータ・ストリームをサポートするプリンターの総称名。

**ibm5575**

5575 漢字プリンター。

**ibm5577**

5577 漢字プリンター。

**注:** テキスト・フォーマット・システムを完全なものにするために、AT&T 流通センターから下記のデバイスがそのままの状態出荷されています。これらのテーブルのサポート機能は用意されていません。

-T Name (続き)

**2631** 正規モードの Hewlett-Packard 2631 プリンター。  
**2631-c** 圧縮モードの Hewlett-Packard 2631 プリンター。  
**2631-e** 拡張モードの Hewlett-Packard 2631 プリンター。  
**300** DASI-300 プリンター。  
**300-12** 1 インチ当たり 12 文字に設定している DASI-300 端末。  
**382** DTC-382。  
**4000a** Trendata 4000a 端末 (4000A)。  
**450** DASI-450 (Diablo Hyterm) プリンター。  
**450-12** 1 インチ当たり 12 文字に設定している DASI-450 端末。  
**832** Anderson Jacobson 832 端末。  
**8510** C.ITOH プリンター。  
**tn300** GE Terminet 300 端末。  
**X** TX 印刷トレイン搭載のプリンター。  
**300s** DASI-300s プリンター (300S)。  
**300s-12**  
1 インチ当たり 12 文字に設定している DASI-300s プリンター (300S-12)。

-u Number

3 番目のフォント位置 (ボールド) のボールド・ファクター (文字重ね打ち数) を指定数値に設定します。Number 変数を指定しない場合は、0 になります。

-z

.tm (ワークステーション・メッセージ) 要求が生成するメッセージのみを出力します。

注: マクロの詳細については、**troff** コマンドのフォーマット・ツールのためのマクロ・パッケージのセクションを参照してください。

-

入力が標準入力から読み込まれるようにします。

## ファイル

**/usr/share/lib/tmac/tmac.\***

標準マクロ・ファイルを指すポインターが入っています。

**/usr/share/lib/macros/\***

標準マクロ・ファイルが入っています。

**/usr/share/lib/nterm/\***

**nroff** コマンド用の端末ドライビング・テーブルが入っています。

**/usr/share/lib/pub/terminals**

サポートされている端末のリストが入っています。

## 関連情報

**col** コマンド、**mm** コマンド、**neqn** コマンド、**tbl** コマンド、**troff** コマンド。

**nroff** および **troff** 入力ファイル・フォーマット。

**troff** コマンドの **nroff** および **troff** コマンド用 **nroff** および **troff** 要求のセクション。

---

## nslookup コマンド

### 目的

インターネット・ドメインのネーム・サーバーを照会します。

## 構文

**nslookup** [ *-Option ...* ] [ *Host* ] [ *-NameServer* ]

## 説明

**nslookup** コマンドは、2つのモードでインターネット・ドメインのネーム・サーバーを照会します。対話モードを使用すると、各種のホストおよびドメインに関する情報についてネーム・サーバーを照会したり、ドメイン内にあるホストの一覧を表示できます。非対話モードでは、指定のホストまたはドメインの名前および要求情報が表示されます。

**nslookup** コマンドが対話モードに入るのは、引数を指定しないとき、あるいは最初の引数が - (負符号) で 2 番目の引数がネーム・サーバーのホスト名または IP アドレスのときです。引数を指定しないと、コマンドはデフォルトのネーム・サーバーを照会します。- (負符号) はオプションのサブコマンド (*-Option...* 変数) を呼び出します。**set** コマンドは例外として、これらのサブコマンドはコマンド行に指定し、**nslookup** コマンドの引数の前になければなりません。代わりに、**set** サブコマンドのオプションをユーザーのホーム・ディレクトリー内の **.nslookuprc** ファイルに指定できます。

**nslookup** コマンドは、最初の引数が検索の対象となるホストの名前または IP アドレスのときには、非対話モードで実行します。この場合、ネーム・サーバーのホスト名または IP アドレスはオプションです。

非対話コマンドは、デフォルトのネーム・サーバーまたは *NameServer* パラメーターで指定したネーム・サーバーを使用して、指定の *Host* に関する情報を探します。*Host* パラメーターに IP アドレスを指定し、照会タイプが **A** または **PTR** の場合は、そのホストの名前が戻されます。*Host* パラメーターに名前を指定し、その名前の末尾にピリオドがない場合は、デフォルトのドメイン名がその名前に追加されます。現行ドメイン内にないホストを探すときには、その名前にピリオドを 1 つ付けます。

注: **set** サブコマンドの **domain**、**srchlist**、**defname**、**search** のオプションは、ユーザーのホーム・ディレクトリーの **.nslookuprc** ファイルに指定すると、非対話式コマンドの動作に影響することがあります。

## サブコマンド

下記のコマンドは、Ctrl-C キー・シーケンスを入力すれば、いつでも割り込むことができます。終了するには、Ctrl-D キー・シーケンスを入力するか、または **exit** とタイプします。組み込みコマンドをホスト名として扱うには、その前にエスケープ文字 (\) を付けます。認識のないコマンドはホスト名と見なされません。

下記のサブコマンドは、**nslookup** コマンドが認識しています。

**finger** [*Name*] [> *FileName*]  
**finger** [*Name*] [>> *FileName*]

現行ホスト上の **finger** デモン・サーバーと接続します。現行ホストは、前回のホスト検索が成功し、**set querytype=A** コマンドで戻されるようなアドレス情報が戻されたときに定義されます。*Name* パラメーターにはユーザー名を指定しますが、これはオプションです。> 文字および >> 文字は、>> 出力を、新規ファイルまたは既存ファイルにリダイレクトするのに使用できます。

**server** *Domain*  
**lserver** *Domain*

デフォルトのサーバーを *Domain* パラメーターで指定した値に変更します。**lserver** サブコマンドは、初期サーバーを使用してドメインに関する情報を探します。**server** サブコマンドは現在のデフォルト・サーバーを使用します。信頼できる回答が見つからない場合には、回答が得られそうな別のサーバーの名前が戻されます。

**root**

デフォルトのサーバーをドメイン名スペースのルートのサーバーに変更します。現時点では、ホスト **ns.nic.ddn.mil** が使用されます。ルート・サーバーの名前は、**set root** サブコマンドで変更できます。( **root** サブコマンドは **lserver ns.nic.ddn.mil** サブコマンドと同じ意味です。)

**ls** [*Option*] *Domain* [**>** *FileName*]  
**ls** [*Option*] *Domain* [**>>** *FileName*]

指定した *Domain* に使用できる情報をリストします。さらにオプションでその出力を *FileName* パラメーターで指定したファイルに追加または作成 (して追加) します。デフォルトの出力には、ホスト名およびそれらの IP アドレスが入ります。*Option* パラメーターに使用できる値は下記のとおりです。

**-t** *QueryType*

指定したタイプのすべてのレコードをリストします。デフォルトのレコード・タイプは **A** です。

**A** ホストの IP アドレス

**CNAME** 別名の正規の名前

**HINFO** ホスト CPU およびオペレーティング・システム・タイプ

**KEY** セキュリティー・キー・レコード

**MINFO** メール・ボックスまたはメール・リスト情報

**MX** メール交換プログラム

**NS** 命名ゾーンのネーム・サーバー

**PTR** 照会が IP アドレスの場合はホスト名、そうでない場合は他の情報を指すポインター

**SIG** 署名レコード

**SOA** ドメインの「権限開始」情報

**TXT** テキスト情報

**UINFO** ユーザー情報

**WKS** サポートされた定式サービス

**-a** ドメイン内のホストの別名をリストします (**-t CNAME** オプションと同じ意味です)。

**-d** ドメインのすべてのレコードをリストします (**-t ANY** オプションと同じ意味です)。

**-h** ドメインの CPU およびオペレーティング・システム情報をリストします (**-t HINFO** オプションと同じ意味です)。

**-s** ドメイン内のホストの定式サービスをリストします (**-t WKS** オプションと同じ意味です)。

注: 出力がファイルにリダイレクトされると、サーバーから 50 レコード受信されるごとにハッシュ・マークが出力されます。

前の **ls** コマンドの出力をソートし、**more** コマンドを使用してそれらをリストします。

**view** *FileName*  
**help**  
**?**  
**exit**

コマンドの簡潔な要約を表示します。  
プログラムを終了します。

**set** *Keyword*[=*Value*]

検索に影響を及ぼす状態情報を変更します。このサブコマンドは、コマンド行に指定することも、オプションでユーザーのホーム・ディレクトリー内の **.nslookuprc** ファイルに入れることもできます。有効なキーワードは下記のとおりです。

**all** 頻繁に使用されるオプションの現在の設定値を表示します。現在のデフォルトのサーバーおよびホストに関する情報も出力されます。

**class=***Value*

照会クラスを下記のいずれかのクラスに変更します。クラスは情報のプロトコル・グループを指定します。デフォルトは **IN** です。

**IN** インターネット・クラス

**CHAOS** カオス・クラス

**HESIOD** MIT Althena Hesiod クラス

**ANY** ワイルドカード (上記のすべて)

**[no]debug**

デバッグ・モードをオンにします。デフォルトは **nodebug** (オフ) です。

**[no]d2**

広範囲のデバッグをオンにします。デフォルトは **nod2** (オフ) です。

**domain=***Name*

デフォルトのドメイン名を *Name* パラメーターで指定した名前に変更します。デフォルトのドメイン名は、**defname** オプションおよび **search** オプションの状態に応じて、検索要求に追加されます。ドメイン検索リストには、検索リストの中にその名前ですくなくとも 2 つのコンポーネントがあれば、デフォルト・ドメインの親が入ります。例えば、デフォルト・ドメインが **CC.Berkeley.EDU** の場合、検索リストは **CC.Berkeley.EDU** および **Berkeley.EDU** となります。別のリストを指定するには、**set srchlist** コマンドを使用します。リストを表示するには、**set all** コマンドを使用します。 **domain=Name** オプションのデフォルトは、システムの **hostname**、**/etc/resolv.conf** または **LOCALDOMAIN** ファイルで指定した値です。

**srchlist=***Name1|Name2|...*

デフォルトのドメイン名を *Name1* パラメーターで指定された名前に、ドメイン検索リストを *Name1*、*Name2*、... パラメーターで指定された名前にそれぞれ変更します。この文字列 (スラッシュで区切られた *Name1*、*Name2*、...、パラメーターを記述) の長さは、1024 文字を超えることはできません。名前を表示するには、**set all** コマンドを使用します。デフォルトは、システムの **hostname**、**/etc/resolv.conf** または **LOCALDOMAIN** ファイルで指定した値です。

注: このコマンドは、**set domain** コマンド・オプションのデフォルトのドメイン名および検索リストをオーバーライドします。

**[no]defname**

デフォルトのドメイン名を単一コンポーネント検索要求 (ピリオドが含まれていない要求) に追加します。デフォルトは **defname** (追加) です。

**[no]search**

検索要求に末尾ピリオド以外のピリオドが含まれている場合、回答が受信されるまで、ドメイン検索リスト内のドメイン名を要求に追加します。デフォルトは **search** です。

**port=***Value*

デフォルトの TCP/UDP ネーム・サーバー・ポートを *Value* パラメーターで指定した番号に変更します。デフォルト値は 53 です。

**querytype=***Value*

**type=Value**

情報照会を下記のいずれかの値に変更します。デフォルトは **A** です。

|              |                                          |
|--------------|------------------------------------------|
| <b>A</b>     | ホストの IP アドレス                             |
| <b>ANY</b>   | 使用可能なすべてのオプション                           |
| <b>CNAME</b> | 別名の正規の名前                                 |
| <b>HINFO</b> | ホスト CPU およびオペレーティング・システム・タイプ             |
| <b>KEY</b>   | セキュリティ・キー・レコード                           |
| <b>MINFO</b> | メール・ボックスまたはメール・リスト情報                     |
| <b>MX</b>    | メール交換プログラム                               |
| <b>NS</b>    | 命名ゾーンのネーム・サーバー                           |
| <b>PTR</b>   | 照会が IP アドレスの場合はホスト名、そうでない場合は他の情報を指すポインター |
| <b>SIG</b>   | 署名レコード                                   |
| <b>SOA</b>   | ドメインの「権限開始」情報                            |
| <b>TXT</b>   | テキスト情報                                   |
| <b>UINFO</b> | ユーザー情報                                   |
| <b>WKS</b>   | サポートされた定式サービス                            |

**[no]recurse**

情報がない場合、ネーム・サーバーに他のサーバーを照会するよう命じます。デフォルトは **recurse** です。

**retry=Number**

要求を再試行する回数を *Number* パラメーターで指定した値に設定します。**set timeout** コマンドで指定した時間枠内に要求に対する応答が受信されないと、タイムアウト期間が 2 倍になり、その要求は再び送信されます。このサブコマンドは、タイムアウトの前に要求を送信する回数を制御します。デフォルト値は 4 です。

**root=Host**

ルート・サーバーの名前を *Host* パラメーターで指定した名前に変更します。デフォルトは **ns.nic.ddn.mil** です。

**timeout=Number**

応答を待つ際の初期タイムアウトの時間間隔を *Number* パラメーターで指定した秒数に変更します。デフォルト値は 5 秒です。

**[no]vc**

要求をサーバーに送信するときに仮想回線を使用します。デフォルトは **novc** (仮想回線なし) です。

**[no]ignoretc**

パケットの打ち切り誤差を無視します。デフォルトは **noignoretc** (無視しない) です。

## 例

1. デフォルトの照会タイプをホスト情報 (**HINFO**) に変更し、初期タイムアウトを 10 秒に変更するには、下記のように入力します。

```
nslookup -query=hinfo -timeout=10
```

2. ドメインと検索リストを 3 つの名前、**1cs.MIT.EDU**、**ai.MIT.EDU**、**MIT.EDU** に設定するには、下記のように入力します。

```
nslookup -set srchlist=1cs.MIT.EDU/ai.MIT.EDU/MIT.EDU
```

このコマンドは、**set domain** コマンドのデフォルトのドメイン名および検索リストをオーバーライドします。リストを表示するには、**set all** コマンドを使用します。

3. 名前がホスト、ドメイン、または他のエンティティーのうちのどれを示すかを定めるには、下記のように入力します。

```
nslookup -querytype=ANY austin.ibm.com
```



**nslookup** コマンドは、下記のように名前 `austin.ibm.com` に関するすべての使用可能な情報 (権限ステートメント (SOA)、ネーム・サーバー、メール交換プログラム、およびホストの IP アドレス情報など) を戻します。

```
Server: benames.austin.ibm.com
Address: 9.3.199.2
```

```
austin.ibm.com origin = ausname1.austin.ibm.com
      mail addr = brian.chriss.austin.ibm.com
      serial=1993081210,refresh=3600,retry=300,expire=604800, min=86400
austin.ibm.com nameserver = ausname1.austin.ibm.com
austin.ibm.com nameserver = bb3names.austin.ibm.com
austin.ibm.com nameserver = benames.austin.ibm.com
austin.ibm.com nameserver = b45names.austin.ibm.com
austin.ibm.com nameserver = bbcnames.austin.ibm.com
austin.ibm.com nameserver = netmail.austin.ibm.com
austin.ibm.com preference = 10, mail exchanger = netmail.austin.ibm.com
austin.ibm.com inet address = 129.35.208.98
ausname1.austin.ibm.com inet address = 129.35.17.2
bb3names.austin.ibm.com inet address = 129.35.208.99
benames.austin.ibm.com inet address = 9.3.199.2
b45names.austin.ibm.com inet address = 129.35.49.2
bbcnames.austin.ibm.com inet address = 129.35.17.68
netmail.austin.ibm.com inet address = 129.35.208.98
```

4. ホスト `opus` 上で非対話式照会を実行するには、下記のように入力します。

```
nslookup opus
```

**nslookup** コマンドは、**host** コマンドと同様に応答します。このコマンドは、下記のようにホスト `opus` のドメイン名および IP アドレスを戻します。

```
Name: opus.austin.ibm.com
Address: 129.35.129.223
```

ホスト `opus` がネーム・サーバー (空の `/etc/resolv.conf` ファイルを指定して **named** デモンを実行するホスト) だった場合には、下記の情報が表示されます。

```
Server: loopback
Address: 0.0.0.0
```

## 終了状況

検索要求が正常に終了しないと、**nslookup** コマンドは下記のエラー・メッセージのいずれかを戻します。

### Timed Out

指定回数の再試行が終わってもサーバーが要求に応答しなかったことを示します。

### No Response from Server

ネーム・サーバーがサーバー・マシンで稼働していないことを示します。

### No Records

ホスト名が有効であっても、そのホストに指定された照会タイプのリソース・レコードをサーバーが持っていないことを示します。

### Non-Existent Domain

ホストまたはドメインの名前が存在しないことを示します。

### Connection Refused

照会の時点でネーム・サーバーまたはフィンガー・サーバーへの接続が行えなかったことを示します。このエラーは通常 **ls** 要求および **finger** 要求に関連付けられます。

### Network Is Unreachable

照会の時点でネーム・サーバーまたはフィンガー・サーバーへの接続が行えなかったことを示します。このエラーは通常 **ls** 要求および **finger** 要求に関連付けられます。

### Server Failure

ネーム・サーバーが内部矛盾を見つけ、有効な回答を戻せなかったことを示します。

|                     |                                                  |
|---------------------|--------------------------------------------------|
| <b>Refused</b>      | ネーム・サーバーが要求に対する応答をリジェクトしたことを示します。                |
| <b>Format Error</b> | 適切なフォーマットでなかったため、ネーム・サーバーが要求パケットをリジェクトしたことを示します。 |

## ファイル

|                           |                                 |
|---------------------------|---------------------------------|
| <b>/usr/bin/nslookup</b>  | <b>nslookup</b> コマンドが入っています。    |
| <b>/etc/resolv.conf</b>   | 初期ドメイン名およびネーム・サーバー・アドレスが入っています。 |
| <b>\$HOME/.nslookuprc</b> | ユーザーの初期オプションが入っています。            |
| <b>HOSTALIASES</b>        | ホストの別名が入っています。                  |
| <b>LOCALDOMAIN</b>        | デフォルトのオーバーライド・ドメインが入っています。      |

## 関連情報

**namerslv** コマンド、**traceroute** コマンド。

**named** デーモン。

**res\_query** サブルーチン、**res\_search** サブルーチン。

TCP/IP 用の **resolv.conf** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージの TCP/IP のネーム・レゾリューションのセクション。

## nsupdate コマンド

### 目的

DNS サーバーを更新します。

### 構文

**nsupdate4**、**nsupdate8**、または **nsupdate9** コマンドの構文を参照してください。

### 説明

AIX は、BIND: 4、8、9 の 3 つのバージョンをサポートします。デフォルトでは、**named** は **named8** に、**nsupdate** は **nsupdate4** に、**named-xfer** は **named-xfer4** にリンクします。別のバージョンの **nsupdate** を使用する場合は、これに応じて **nsupdate** コマンドのシンボリック・リンクを再リンクする必要があります。

例えば **nsupdate9** を使用するには、次のように入力します。

```
ln -fs /usr/sbin/nsupdate9 /usr/sbin/nsupdate
```

**nsupdate4** は **named8** で使用できますが、**nsupdate9** はセキュリティー処理が異なるため、**named9** で使用する必要があります。

## ファイル

### `/usr/sbin/named`

システム上で使用される **named** のバージョンへのシンボリック・リンクが入っています。

### `/usr/sbin/nsupdate`

システム上で使用される **nsupdate** のバージョンへのシンボリック・リンクが入っています。

### `/usr/sbin/nsupdate4`

BIND バージョン 4 **nsupdate** コマンドが入っています。

### `/usr/sbin/nsupdate8`

BIND バージョン 8 **nsupdate** コマンドが入っています。

### `/usr/sbin/nsupdate9`

BIND バージョン 9 **nsupdate** コマンドが入っています。

## 関連情報

**nsupdate4** コマンド、 **nsupdate8** コマンド、 **nsupdate9** コマンド。

**named** デーモン。

**bootp** 構成ファイル、 **DHCP Client** 構成ファイル、 **DHCP Server** 構成ファイル。

ネットワークおよびコミュニケーションのマネージ のネーム・サーバー・レゾリューションのセクション、およびドメイン・ネーム・レゾリューションの計画のセクション。

ネットワークおよびコミュニケーションのマネージ の TCP/IP アドレスとパラメーターの割り当て - 動的ホスト構成プロトコル (DHCP) のセクション。

ネットワークおよびコミュニケーションのマネージ の TCP/IP のネーム・レゾリューションのセクション、および TCP/IP デーモンのセクション。

---

## nsupdate4 コマンド

### 目的

DNS サーバーを更新します。

### 構文

```
nsupdate4 [ -a ] [ -g ] [ -i ] [ -q ] [ -v ] [ -? ] [ -k KeyFile ] [ -h HostName ] [ -d DomainName ]  
[ -p PrimaryName ] [ -r IPAddress ] [ -s "CommandString" ]
```

### 説明

**nsupdate4** コマンドは DNS サーバーを更新します。**nsupdate4** コマンドは、対話モードまたはコマンド・モードで実行します。コマンド・ストリングが提供された場合、**nsupdate4** コマンドは、コマンド・ストリングを実行してから終了します。リターン・コードは、コマンド・ストリングの成功に依存していません。

コマンド・ストリングまたは対話モードに有効な内部コマンドは、下記のとおりです。

r       更新パケットをリセットします。このコマンドは、最初でなければなりません。

|   |                                                                  |
|---|------------------------------------------------------------------|
| d | レコードを削除します。このコマンドの後、レコード・タイプおよび削除する値の質問があります。                    |
| a | レコードを追加します。このコマンドの後、レコード・タイプおよび追加する値の質問があります。                    |
| n | レコードがまだ存在していない場合のみ、レコードを追加します。このコマンドの後、レコード・タイプおよび追加する値の質問があります。 |
| e | レコードが既に存在している場合のみ、レコードを追加します。このコマンドの後、レコード・タイプおよび追加する値の質問があります。  |
| t | 更新済みレコードに値を示すデフォルトの時刻を設定します。                                     |
| s | 更新に符号を付けます。-a または -g フラグが指定されたかどうかにより、キーが生成され、更新に符号が付けられます。      |
| x | 更新パケットを -p フラグで指定したサーバーに送信します。                                   |
| v | 詳細モードをオンまたはオフにします。                                               |
| i | パラメーターによって渡された情報を戻します。                                           |
| p | レコード・フォーマットで更新パケットを表示します。                                        |
| q | コマンドを終了します。                                                      |

**-g** フラグを使用すれば、セキュア・モードで使用するためにクライアントに配布する鍵のセットを生成することができます。このフラグは、ホスト名および基本名を取って、公開鍵および秘密鍵を生成します。セキュア・モード・ゾーン操作では、公開鍵は保護するデータの DNS サーバーのデータベースに入れられませんが、秘密鍵はクライアント上に置かれるので、後でその情報を更新することができます。

**-a** フラグを使用すれば、管理モードに入ることができます。このゾーンは、ゾーン・キーで保護することができます。このキーを使用すると、ユーザーはゾーンにフル・アクセスすることができます。**-a** フラグは、個々のレコード・キーではなく、更新署名用のゾーン・キーを使用しようとしています。

## フラグ

|                                  |                                                      |
|----------------------------------|------------------------------------------------------|
| <b>-a</b>                        | 管理モードです。個々のレコード・キーではなくゾーン・キーを使用しようとしています。            |
| <b>-d</b> <i>DomainName</i>      | 更新を適用するドメインの名前を指定します。これは、PTR レコード以外のすべてのレコードで使用されます。 |
| <b>-g</b>                        | 生成モードです。基本名およびホスト名用の 1 対の鍵を生成するのに使用されません。            |
| <b>-h</b> <i>HostName</i>        | 更新するレコードの名前を指定します。これは、PTR レコード以外のすべてのレコードで使用されます。    |
| <b>-i</b>                        | エラーを無視して、文字列のすべてのコマンドを実行します。                         |
| <b>-k</b> <i>KeyFile</i>         | デフォルトのキー・ファイルの名前を指定します。これは、キー用のファイルです。               |
| <b>-p</b> <i>PrimaryName</i>     | DNS サーバーの名前または IP アドレスを指定します。1 次 DNS サーバーが推奨されます。    |
| <b>-q</b>                        | 出力をオフにします。                                           |
| <b>-r</b> <i>IPAddress</i>       | 更新するレコードの IP アドレスを指定します。これは、PTR レコードのみで使用されます。       |
| <b>-s</b> <i>"CommandString"</i> | スペースまたはコロンで区切られた内部コマンドのセットです。                        |
| <b>-v</b>                        | 詳細出力をオンにします。                                         |
| <b>-?</b>                        | コマンド行オプション・リストです。                                    |

## 終了状況

このコマンドは、以下の終了値を戻します。

|    |             |
|----|-------------|
| 0  | 正常終了。       |
| >0 | エラーが発生しました。 |

## セキュリティ

アクセス制御: 任意のユーザー

### 例

パケットを初期化し、指定されたホスト名のすべての A レコードを削除し、3110400 のデフォルトのキーパッドで符号が付けられ、300 秒間有効な 9.3.145.2 アソシエーションへそのホスト名の 1 つの A レコードを追加し、パケットを送信し、終了するには、下記のように入力します ("," の箇所では改行キーを押します)。

```
r;d;a;*;a;a;9.3.145.2;s;300;3110400;x;q
```

エントリーに 1 つでも失敗した場合、メッセージが表示されます。コマンド行モードでは、エラーが発生するとプログラムが終了して 1 を戻します。

## ファイル

`/usr/sbin/nsupdate4`

`nsupdate4` コマンドが入っています。

`/usr/sbin/named`

DNS サーバーが入っています。

## 関連情報

DHCP Client Configuration File

DHCP Server Configuration File

bootp Configuration File

ネットワークおよびコミュニケーションのマネージの TCP/IP アドレスとパラメーターの割り当て - 動的ホスト構成プロトコル (DHCP) のセクション。

ネットワークおよびコミュニケーションのマネージの TCP/IP デーモンのセクション。

---

## nsupdate8 コマンド

### 目的

BIND 8 ネーム・サーバーが読める DNS 更新パケットを生成します。

### 構文

```
nsupdate8 [ -v ] [ -d ] [Filename]
```

### 説明

`nsupdate8` コマンドは、コマンド行に指定されたファイルから、パイプ、またはファイルからリダイレクトされた入力の `stdin` から、または `tty` から対話式に読み取ることができます。これらの 3 つの方法では、下記に指定する同じフォーマットを使用します。入力は、ZONE を更新するために使用できる DNS

更新パケットを定義します。更新には、前提条件セクションと更新セクションの 2 つのセクションがあります。DNS ネーム・サーバーは、更新セクションを処理する前にすべての前提条件が真であるか検証します。

## フラグ

- d** **nsupdate8** が、そのアクションに関する追加のデバッグ情報を生成するようにします。
- v** **nsupdate8** に対して、通常の UDP 接続ではなく、仮想回線 (TCP 接続) を使用するように指示します。

入力形式は、1 組の更新パケットとして定義されます。各パケットは、改行で終了される文字列セットです。入力ストリーム内の最後の文字列は、EOF によって終了する場合があります。そのストリームに複数の更新パケットが含まれる場合、各パケットは空白行 (単一の改行文字) によって次のパケットと区切る必要があります。セミコロンはコメント文字として使用されます。それ以降の内容はすべて無視され、更新パケットから出されます。

**nsupdate8** の入力形式は次のとおりです。

```
section opcode name [ttl] [class] [type] [data]
```

これは一般的な形式です。*section* と *opcode* の各値によって、それ以降の引数に必要な内容が修正されます。

*section*                   このレコードの対象となる更新のセクションを定義します。値は次のとおりです。

**prereq**   そのレコードが前提条件セクションに関するものであることを示します。

**update**   そのレコードが更新セクションに関するものであることを示します。

*opcode*

このレコードを使用して行う操作を定義します。

**値:** 前提条件操作:

**nxdomain**

その名前が存在していないか確認することを示します。 *ttl* にはゼロ以外の値を指定して、その名前が存在してはならない期間を示す必要があります。オプション・クラスを指定して、検索をそのクラスだけに限定できます。 *T\_ANY* のタイプは、どのレコード・タイプとも一致するようにワイルドカードとして使用されます。

**nydomain**

その名前が存在しているか確認することを示します。*ttl* にはゼロ以外の値を指定して、その名前が存在し続けなければならない期間を示す必要があります。オプション・クラスでは、検索をそのクラスだけに限定できます。レコード・タイプは *T\_NONE* です。これは、名前が存在することを確かめる検査を強制します。

**nrrrset** その名前について、特定タイプのレコードが存在しないことを示します。オプションのクラスと *ttl* は、検索を制限することができます。タイプは必須です。

**nyrrrset** その名前について、特定タイプのレコードが存在しなければならないことを示します。オプションの *ttl* およびクラスは、検索を制限します。タイプおよびデータは必須です。データはワイルドカードの場合があります。データがワイルドカードでない場合、そのデータは、指定されたタイプのフォーマットと一致していなければなりません。

**値:** 更新操作:

**add** レコードをそのゾーンに追加することを示します。タイプおよびデータは必須です。ワイルドカードはデータとしては認められません。*ttl* は必須で、ゼロ以外でなければなりません。クラスはオプションです。

**delete** レコードをそのゾーンから削除することを示します。タイプおよびデータはオプションです。ワイルドカードはデータとして認められます。データは、NULL 文字列をデフォルトとしてとり、タイプはデフォルトとして *T\_ANY* をとります。*ttl* およびクラスはオプションです。*ttl* は、指定されると 0 にリセットされます。

*name*

テストまたは修正が行われている DNS エントリーの名前。

[*ttl*]

追加されているレコードのオプションの存続時間。形式によっては、これがオプションでない場合があります。

[*class*]

ゾーンに追加されるレコードのクラス。値は *IN*、*HESIOD*、および *CHAOS* です。メッセージ全部のデフォルトは *IN* です。

[*type*]

ゾーンに追加されるレコード、またはゾーンと突き合わされるレコードのタイプ。値は、*A*、*NS*、*CNAME*、*SOA*、*MB*、*MR*、*NULL*、*WKS*、*PTR*、*HINFO*、*MINFO*、*MX*、*TXT*、*RP*、*AFSDB*、*X25*、*ISDN*、*RT*、*NSAP*、*NSAP\_PTR*、*PX*、および *LOC* です。注: *CNAME* タイプは、*TSIG* および *TKEY* レコードの場合にのみ追加できます。これらのレコードは現在 *BIND 8* ではサポートされていません。

[*data*]

ゾーンに追加されるデータ、またはゾーンと突き合わされるデータ。このデータは、指定されたタイプについて、および DNS サーバー・ゾーン・ファイルの *DOMAIN* データ・ファイル内で有効でなければなりません。前提条件の検査では、どの値とも一致するようにアスタリスク (\*) が使用されます。これは、特定タイプの全レコードを削除する場合にも使用できます。

以下に、特定のフォーマットの場合を示します。



```
prereq nxdomain <name> <ttl != 0> [class]
prereq nydomain <name ttl != 0> [class]
prereq nxrrset <name> [ttl] [class] <type>
prereq nyrrset <name> [ttl] [class] <type> <data>
update delete <name> [ttl] [class] [type] [data]
update add <name> <ttl != 0> [class] <type> <data>
```

## 診断

実行されたさまざまなアクションまたはプログラムが検出した問題 (あるいはその両方) を示すメッセージ。

## 関連情報

**nsupdate** コマンド、**named** コマンド。

**named.conf** ファイル・フォーマット、**DOMAIN Cache** ファイル・フォーマット、**DOMAIN Data** ファイル・フォーマット、**DOMAIN Reverse Data** ファイル・フォーマット、**DOMAIN Local Data** ファイル・フォーマット、**resolv.conf** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ の TCP/IP のネーム・レゾリューションのセクション、および TCP/IP デーモンのセクション。

ネットワークおよびコミュニケーションのマネージ のネーム・サーバー・レゾリューションのセクション、およびドメイン・ネーム・レゾリューションの計画のセクション。

---

## nsupdate9 コマンド

### 目的

動的 DNS 更新ユーティリティ。

### 構文

```
nsupdate9 [-d] [-y keyname:secret | -k keyfile] [-v] [filename]
```

### 説明

**nsupdate9** コマンドは、RFC2136 で定義された動的 DNS 更新要求をネーム・サーバーにサブミットします。これにより、ゾーン・ファイルを手作業で編集しなくても、リソース・レコードをゾーンに追加または除去することができます。1 つの更新要求に、複数のリソース・レコードの追加または除去の要求を含めることができます。

**nsupdate9** または DHCP サーバーを使用した動的制御下にあるゾーンは、手作業で編集しないでください。手作業による編集は動的更新と矛盾する可能性があり、データ破損の原因になります。

**nsupdate9** を使用して動的に追加または除去するリソース・レコードは、同じゾーンにある必要があります。要求はゾーンのマスター・サーバーに送信されます。これは、ゾーンの SOA レコードの MNAME フィールドによって指定されます。

**-d** オプションは、**nsupdate9** をデバッグ・モードにします。これは、行われる更新要求とネーム・サーバーから受信する応答のトレース情報を提供します。

動的 DNS 更新を認証するために、トランザクション署名を使用することができます。これは RFC2845 に記述されている TSIG リソース・レコード・タイプを使用します。署名は、**nsupdate9** とネーム・サーバーだけが知っている共有秘密に依存しています。現在、TSIG 用にサポートされる暗号化アルゴリズムは HMAC-MD5 (RFC 2104 で定義されたもの) だけです。TSIG 用に他のアルゴリズムも定義されるようになれば、アプリケーションはお互いを認証する時に、キーだけでなく、該当のアルゴリズムを選択していることを確認する必要があります。例えば、適切なキーおよびサーバー・ステートメントを **/etc/named.conf** に追加し、ネーム・サーバーが該当の秘密鍵とアルゴリズムを、TSIG 認証を使用するクライアント・アプリケーションの IP アドレスと関連付けることができるようにします。**nsupdate9** は **/etc/named.conf** を読み取りません。

**nsupdate9** は **-y** または **-k** オプションを使用して、動的 DNS 更新要求を認証するための TSIG レコードの生成に必要な共有秘密を提供します。これらのオプションはどちらか一方を使用します (両方を指定することはできません)。**-k** オプションの場合、**nsupdate9** はファイル *keyfile* (この名前の形式は **K{name}+.157.+[random].private**) から共有秘密を読み取ります。歴史的な理由から、ファイル **K{name}+.157.+[random].key** も存在する必要があります。**-y** オプションを使用すると、*keyname:secret* から署名が生成されます。*keyname* はキーの名前であり、*secret* は base64 でエンコードされた共有秘密です。**-y** オプションを使用することはお勧めできませんが、その理由は、共用 *secret* がコマンド行引数として平文で提供されるためです。これは ps(1) からの出力で、またはユーザーのシェルで保守されるヒストリー・ファイルで見ることができます。

デフォルトでは、**nsupdate9** は UDP を使用してネーム・サーバーに更新要求を送信します。**-v** オプションは **nsupdate9** が TCP 接続を使用するようにします。これは、更新要求をまとめて行う場合には望ましい方法です。

## フラグ

**-d** **nsupdate9** をデバッグ・モードで操作する。  
**-k** *keyfile* ファイル *keyfile* から共有秘密を読み取る。  
**-v** **nsupdate9** が TCP 接続を使用するようにする。  
**-y** *keyname:secret* から署名を生成する。  
*keyname:secret*

## パラメーター

**filename** 更新されるファイル。

## 入力フォーマット

**nsupdate9** は、ファイル *filename* または標準入力から入力を読み取ります。それぞれのコマンドは、入力の 1 行だけで提供します。いくつかのコマンドは管理目的用です。他のコマンドは、更新命令またはゾーンの内容についての前提条件の検査です。これらの検査は、ある名前またはリソース・レコードのセット (RRset) がゾーンに存在する、または欠落している、といった条件をセットします。更新要求全体を先に進めるには、これらの条件を満たす必要があります。前提条件のテストを満たさないと、更新は拒否されません。

どの更新要求も、ゼロ以上の前提条件と、ゼロ以上の更新からなります。これにより、指定した何らかのリソース・レコードがゾーンに存在する、またはゾーンに存在しない場合、認証された更新要求を適切に先に進めることができます。ブランクの入力行 (または *send* コマンド) は、累積したコマンドを 1 つの動的 DNS 更新要求としてネーム・サーバーに送信します。

コマンド・フォーマットと意味は、次のとおりです。

|                                                                                                                   |                                                                                                                                                                                                                                                                                        |
|-------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>server</b> [ <i>servername</i> ] [ <i>port</i> ]                                                               | すべての動的更新要求をネーム・サーバー <i>servername</i> に送信します。 <b>server</b> ステートメントを指定しないと、 <b>nsupdate9</b> は更新を正しいゾーンのマスター・サーバーに送信します。そのゾーンの SOA レコードの MNAME フィールドは、そのゾーンのマスター・サーバーを示します。 <i>port</i> は、動的更新要求が送信される <i>servername</i> 上のポート番号です。 <i>port</i> 番号を指定しないと、デフォルト DNS ポート番号の 53 が使用されます。 |
| <b>local</b> [ <i>address</i> ] [ <i>port</i> ]                                                                   | すべての動的更新要求をローカル・アドレスを使用して送信します。 <b>local</b> ステートメントを指定しないと、 <b>nsupdate9</b> はシステムが選択した <i>address</i> と <i>port</i> を使用して更新を送信します。 <i>port</i> は、特定のポートから来る要求の作成にも使用できます。ポート番号を指定しないと、システムがポート番号を割り当てます。                                                                             |
| <b>zone</b> [ <i>zonename</i> ]                                                                                   | すべての更新をゾーン <i>zonename</i> に行うことを指定します。 <b>zone</b> ステートメントを指定しないと、 <b>nsupdate9</b> は残りの入力に基づいて、更新する正しいゾーンを判別しようとします。                                                                                                                                                                 |
| <b>key</b> [ <i>name</i> ] [ <i>secret</i> ]                                                                      | すべての更新が <i>keyname keysecret</i> の対を使用した TSIG 署名方式であることを指定します。 <b>key</b> コマンドは、 <b>-y</b> または <b>-k</b> を使用して、コマンド行に指定されたすべてのキーをオーバーライドします。                                                                                                                                           |
| <b>prereq nxdomain</b><br>[ <i>domain-name</i> ]                                                                  | 名前 <i>domain-name</i> を持つ、どのタイプのリソース・レコードも存在しないことを前提条件とします。                                                                                                                                                                                                                            |
| <b>prereq yxdomain</b><br>[ <i>domain-name</i> ]                                                                  | <i>domain-name</i> が存在することを前提条件とします (何らかのタイプの、少なくとも 1 つのリソース・レコードがあること)。                                                                                                                                                                                                               |
| <b>prereq nxrrset</b> [ <i>domain-name</i> ]<br>[ <i>class</i> ] [ <i>type</i> ]                                  | 指定した <i>type</i> 、 <i>class</i> 、および <i>domain-name</i> のリソース・レコードが 1 つも存在しないことを前提条件とします。 <i>class</i> を省略すると IN (インターネット) が想定されます。                                                                                                                                                    |
| <b>prereq yxrrset</b> [ <i>domain-name</i> ]<br>[ <i>class</i> ] [ <i>type</i> ]                                  | これは、指定した <i>type</i> 、 <i>class</i> および <i>domain-name</i> のリソース・レコードが存在しなければならないことを前提条件とします。 <i>class</i> を省略すると IN (インターネット) が想定されます。                                                                                                                                                |
| <b>prereq yxrrset</b> [ <i>domain-name</i> ]<br>[ <i>class</i> ] [ <i>type</i> ] [ <i>data...</i> ]               | 共通の <i>type</i> 、 <i>class</i> 、および <i>domain-name</i> を共用する、この形式の前提条件のセットからのデータを組み合わせて、RR のセットが作られます。RR のこのセットは、指定された <i>type</i> 、 <i>class</i> 、および <i>domain-name</i> でゾーンに存在する RR のセットと、正確に一致する必要があります。 <i>data</i> はリソース・レコードの RDATA の標準テキスト表現で書き込まれます。                          |
| <b>update delete</b> [ <i>domain-name</i> ]<br>[ <i>ttl</i> ] [ <i>class</i> ] [ <i>type</i> ] [ <i>data...</i> ] | <i>domain-name</i> という名前のすべてのリソース・レコードを削除します。 <i>type</i> と <i>data</i> を指定すると、一致したリソース・レコードだけが除去されます。クラスを指定しないと、インターネット <i>class</i> が想定されます。 <i>ttl</i> は無視され、互換性目的でのみ許容されます。                                                                                                        |
| <b>update add</b> [ <i>domain-name</i> ]<br>[ <i>ttl</i> ] [ <i>class</i> ] [ <i>type</i> ] [ <i>data...</i> ]    | 指定された <i>ttl</i> 、 <i>class</i> および <i>data</i> を持つ、新規リソース・レコードを追加します。                                                                                                                                                                                                                 |
| <b>show</b>                                                                                                       | 最後の送信以後に指定された、すべての前提条件および更新を含む、現行メッセージを表示します。                                                                                                                                                                                                                                          |
| <b>send</b>                                                                                                       | 現行メッセージを送信します。これは空白行の入力と同等です。                                                                                                                                                                                                                                                          |

セミコロンで始まる行はコメントであり、無視されます。

## 例

**注:** **nsupdate9** コマンドは、1 つの更新に結合された 2 つの更新を、異なるゾーンにソートしません。2 つの更新は、空白行を挿入するか、**send** コマンドをその 2 つの更新の間に挿入することにより、個別に実行する必要があります。

この例は、**nsupdate9** を使用して `example.com` zone にリソース・レコードを挿入、削除する方法を示しています。それぞれの例の入力は、末尾に空白行を含み、これによりコマンドのグループを 1 つの動的な更新要求として `example.com` のマスター・ネーム・サーバーに送信していることに注目してください。

```
# nsupdate9
> update delete oldhost.example.com A
> update add newhost.example.com 86400 A 172.16.1.1
>
```

`oldhost.example.com` のすべての A レコードが削除されます。また、IP アドレス `172.16.1.1` の `newhost.example.com` の A レコードが追加されます。新しく追加されたレコードは 1 日の TTL (86400 秒) を持ちます。

```
# nsupdate9
> prereq nxdomain nickname.example.com
> update add nickname.example.com CNAME somehost.example.com
>
```

前提条件はネーム・サーバーを入手し、**nickname.example.com** について、どのようなタイプのリソース・レコードも存在しないことを検査します。存在すると、更新要求は失敗します。この名前が存在しない場合は、その CNAME が追加されます。これにより、CNAME の追加時に、ある名前が CNAME として存在する場合、これと同じ名前の他のレコード・タイプが存在してはならないという、RFC1034 の以前からの規則を守ることが保証されます。(この規則は RFC2535 の DNSSEC について、CNAME が SIG、KEY および NXT レコード番号を持てるように変更されました。)

```
# nsupdate9
> update delete 61.26.31.9.in-addr.arpa 0 IN PTR
> update add 61.26.31.9.in-addr.arpa 86400 IN PTR newhost.example.com.
```

IP アドレス `9.31.26.61` の PTR レコードはすべて削除され、IP アドレス `9.31.26.61` およびホスト名 **newhost.example.com** の PTR レコードが追加されます。新しく追加されたレコードは、1 日の TTL (86400 秒) を有します。

## ファイル

|                                             |                                                               |
|---------------------------------------------|---------------------------------------------------------------|
| <code>/etc/resolv.conf</code>               | デフォルト・ネーム・サーバーを識別します。                                         |
| <code>K{name}+.157.#{random}.key</code>     | <b>dnssec-keygen</b> (8) により作成される HMAC-MD5 キーの Base-64 エンコード。 |
| <code>K{name}+.157.#{random}.private</code> | <b>dnssec-keygen</b> (8) により作成される HMAC-MD5 キーの Base-64 エンコード。 |

## 関連情報

RFC2136、RFC3007、RFC2104、RFC2845、RFC1034、および RFC2535。

**named9** デーモンおよび **dnssec-keygen** コマンド。

---

## ntpdate コマンド

### 目的

Network Time Protocol (NTP) を使用して日時を設定します。このコマンドは、AIX 4.2 以降にだけ適用されます。

### 構文

```
ntpdate [ -b ] [ -c ] [ -d ] [ -s ] [ -u ] [ -a Keyid ] [ -e AuthenticationDelay ] [ -k KeyFile ] [ -o Version ] [ -p Samples ] [ -t TimeOut ] Server ...
```

### 説明

**ntpdate** コマンドは、正確な時刻を決定するために指定された NTP サーバーをポーリングして、現地日時を設定します。指定されたそれぞれのサーバーから数多くのサンプルを獲得し、標準 NTP クロック・フィルターおよび選択アルゴリズムを使用して、最良のサンプルを選択します。

**ntpdate** コマンドは、下記のいずれかの方法で時間調整を行います。

- クロックが 0.5 秒より長い間オフになっていると判別すると、**settimeofday** サブルーチンを呼び出してクロックの時間を進めます。これは、ブート時に推奨される方法です。
- クロックが 0.5 秒よりも短い間オフになっていると判別すると、オフセットで **adjtime** サブルーチンを呼び出してクロックの時間を遅らせます。この方法は、安定度のある程度欠きますが、正しく進まないクロックをより正確に保つ傾向があります。デーモンではなく、**cron** コマンドから定時ベースの **ntpdate** コマンドを実行するときは、1 時間または 2 時間ごとに行えば、十分正確に時を刻むことになるのでクロックが進むのを避けることができます。

#### 注:

1. **ntpdate** コマンドの信頼性および精度は、サーバーの数が増えると劇的に改善されます。ユーザーは単一サーバーを使用できますが、少なくとも 3 つまたは 4 つのサーバーを提供することで、より優れたパフォーマンスが得られます。
2. **xntpd** デーモンのような NTP サーバー・デーモンが同一ホスト上で稼働している場合、**ntpdate** コマンドは日付の設定をリジェクトします。
3. このコマンドを実行するには、ローカル・ホスト上で root 権限を持っていないければなりません。

### フラグ

|                                      |                                                                                                                     |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------|
| <b>-a</b> <i>Keyid</i>               | 認証機能を使用可能にし、 <i>Keyid</i> を使用してすべてのパケットを認証します。デフォルトでは、認証機能は使用不可です。                                                  |
| <b>-b</b>                            | <b>settimeofday</b> サブルーチンを呼び出して、クロックの時間を進めます。                                                                      |
| <b>-c</b>                            | <b>adjtime</b> サブルーチンを呼び出して、クロックの時間を進めます。                                                                           |
| <b>-d</b>                            | デバッグ・モードを指定します。 <b>ntpdate</b> コマンドがどのような結果をもたらすかを、実行せずに判別します。結果は、画面上に表示されます。このフラグは、非権限ポートを使用します。                   |
| <b>-e</b> <i>AuthenticationDelay</i> | 認証処理を遅らせる時間量を秒数で指定します。標準的な値の範囲は 0.0001 から 0.003 です。                                                                 |
| <b>-k</b> <i>KeyFile</i>             | デフォルトの <b>/etc/ntp.keys</b> ファイルを使用しないときに、キーが入っているファイルに別の名前を指定します。 <i>KeyFile</i> の説明は、 <b>xntpd</b> コマンドを参照してください。 |

|                          |                                                                                                                                                      |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-o</b> <i>Version</i> | 出力パケットをポーリングするときに使用する NTP バージョン・インプリメンテーションを指定します。 <i>Version</i> の値には、1、2 または 3 を指定できます。デフォルト値は 3 です。                                               |
| <b>-p</b> <i>Samples</i> | 各サーバーから獲得するサンプルの数を指定します。 <i>Samples</i> の値には、1 から 8 までの値を指定できます。デフォルト値は 4 です。                                                                        |
| <b>-s</b>                | 標準出力の使用ではなく、アクションのログを記録するシステム・ログ機能の使用を指定します。 <b>cron</b> コマンドを使用して <b>ntpdate</b> コマンドを実行するときに役に立ちます。                                                |
| <b>-t</b> <i>TimeOut</i> | 応答の待ち時間量を指定します。 <i>TimeOut</i> に指定する値は、0.2 秒の倍数に丸められます。デフォルトは 1 秒です。                                                                                 |
| <b>-u</b>                | パケットの送信元の特権ポートの使用を指定します。特権ポートへの着信通信を覆うファイアウォールの内側において、そのファイアウォールの外側のホストと同期化したいときに役に立ちます。ファイアウォールとは、外部ネットワークから専用ネットワークへのアクセスを制御するシステムまたはコンピューターのことです。 |

## パラメーター

*Server...*                      ポーリングするサーバーを指定します。

## 終了状況

このコマンドは、以下の終了値を戻します。

0        正常終了。  
>0      エラーが発生しました。

## セキュリティ

アクセス制御: このコマンドを実行するには、root 権限を持っていないければなりません。

イベントの監査: N/A

## 例

アドレス 9.3.149.107 の NTP サーバーをポーリングして現地日時を設定するには、下記のように入力します。

```
/usr/sbin/ntpdate 9.3.149.107
```

下記のような出力が表示されます。

```
28 Feb 12:09:13 ntpdate [18450]: step time server 9.3.149.107
offset 38.417792 sec
```

## ファイル

|                          |                             |
|--------------------------|-----------------------------|
| <b>/usr/sbin/ntpdate</b> | <b>ntpdate</b> コマンドが入っています。 |
| <b>/etc/ntp.keys</b>     | デフォルトのキー・ファイルが入っています。       |



## 関連情報

コマンド: **ntpq**、**ntptrace**、**xntpdc**

デーモン: **xntpd**

---

## ntpq コマンド

### 目的

標準の Network Time Protocol (NTP) 照会プログラムを始動します。このコマンドは、AIX 4.2 以降にだけ適用されます。

### 構文

```
ntpq [ -i ] [ -n ] [ -p ] [ -c SubCommand] [ Host ... ]
```

### 説明

**ntpq** コマンドは、指定されたホスト上で実行する NTP サーバーに照会します。そのホストは、現行状態に関する推奨 NTP モード 6 の制御メッセージ形式をインプリメントし、しかもその状態に変更を要求することができます。対話モードでもコマンド行引数を使用しても実行できます。任意の変数の読み取りおよび書き込みを要求することができ、さらに未フォーマット化出力オプションおよびフォーマット化された出力オプションを使用できます。また、**ntpq** コマンドは、サーバーに複数の照会を送信して、共通フォーマットでピアのリストを獲得し表示することもできます。

1 つまたは複数のフラグを持つ **ntpq** コマンドを入力した場合、それぞれの指定された (またはデフォルトがローカル・ホストの) ホスト上で実行する NTP サーバーが、それぞれの要求を受信します。フラグを何も入力しないと、**ntpq** コマンドは、最初に指定されたホストまたはデフォルトのローカル・ホストで実行する NTP サーバー上で、標準入力からコマンドを読み取って実行しようとしています。標準入力端末である場合は、サブコマンドにプロンプトを指示します。

**ntpq** コマンドは、NTP モード 6 のパケットを使用して NTP サーバーと通信し、互換性のあるサーバーを使用できるネットワーク上でそれに照会することができます。

**ntpq** コマンドは、要求の再送を一度試みますが、リモート・ホストが適切な時間内に応答しない場合には要求はタイムアウトになります。

**-i** または **-n** 以外のフラグを指定すると、指定されたホストにすぐに照会が送信されます。そうでなければ、**ntpq** コマンドは、標準入力から対話式フォーマット・サブコマンドを読み取ろうとします。

### フラグ

- c SubCommand** 対話式フォーマットのコマンドを指定します。このフラグは、指定されたホスト上で実行するコマンドのリストに *SubCommand* を追加します。複数の **-c** フラグを入力できます。
- i** 対話モードを指定します。標準出力でプロンプトを表示し、標準入力力でコマンドを読み取ります。
- n** 正規ホスト名ではなくドット 10 進数 (x.x.x.x) ですべてのホスト・アドレスを表示します。
- p** サーバーが認識するピアのリストおよびそれらの状態の要約を表示します。**peers** サブコマンドを使用するのと同じことです。



## パラメーター

*Host ...* ホストを指定します。

## 終了状況

このコマンドは、以下の終了値を戻します。

0 正常終了。  
>0 エラーが発生しました。

## セキュリティ

アクセス制御: このコマンドを実行するには、システム・グループの一部でなければなりません。

イベントの監査: N/A

## 例

1. 対話モードで Network Time Protocol の照会プログラムを開始するには、下記のように入力します。

```
ntpq -i
```

2. 1000 ミリ秒の時間間隔をタイム・スタンプに追加するには、下記のように入力します。

```
ntpq -c "delay 1000"
```

## ntpq 内部サブコマンド

下記のサブコマンドは、**ntpq** 照会プログラムの実行中のみ使用できます。

### 対話式フォーマット・サブコマンド

対話式フォーマット・サブコマンドは、0 個から 4 個までの引数が後に付いたキーワードで構成されます。一意的にサブコマンドを識別するには、全キーワードのうち必要な文字しか入力する必要はありません。サブコマンドの出力は標準出力に表示されますが、個々のサブコマンドの出力は、> (「より大」符号)の後にファイル名を付けてコマンド行に付加し、ファイルにリダイレクトすることができます。

対話式フォーマット・サブコマンドには、完全に **ntpq** 照会プログラム内だけで実行するものや、NTP モード 6 の要求をサーバーに送信しないものがいくつかあります。

NTP モード 6 のメッセージで送られるデータは、下記のフォーマットのエントリー・リストで構成されます。

*Variable=Value*

ここでは、変数を読み取るサーバーへの要求に応じて、*Value* は無視されるので、省略することができます。**ntpq** 照会プログラムは、内部リストを保持しますが、そのリスト内では **readlist** および **writelist** 制御メッセージ・サブコマンドを使用して、制御メッセージに組み込まれるデータをアSEMBルし送信することができます。

? [ *SubCommand* ]

コマンド使用情報を表示します。*SubCommand* を指定しないで使用すると、すべての **ntpq** コマンド・キーワードのリストが表示されます。*SubCommand* を指定して使用すると、サブコマンドに関する機能および使用情報が表示されます。

|                                                            |                                                                                                                                                                                                      |
|------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>addvars</b> <i>Variable</i> [ = <i>Value</i> ] [ ,... ] | 内部データ・リストに追加される変数およびそのオプション値を指定します。複数の変数を追加する場合、リストをコンマで区切り、しかもスペースが入らないようにしなければなりません。                                                                                                               |
| <b>authenticate</b> <b>yes</b>   <b>no</b>                 | すべての要求に関する認証を送信するかどうかを指定します。通常、 <b>ntpq</b> 照会プログラムは、書き込み要求ではない限り、要求を認証しません。                                                                                                                         |
| <b>clearvars</b>                                           | すべての変数を内部データ・リストから除去します。                                                                                                                                                                             |
| <b>cooked</b>                                              | リモート・サーバーから受信したすべての結果を再フォーマットして表示します。末尾の ? (疑問符) は、デコード可能な値を持たない変数にマークを付けます。                                                                                                                         |
| <b>debug</b> <b>more</b>   <b>less</b>   <b>off</b>        | <b>ntpq</b> 照会プログラムのデバッグをオンまたはオフにします。 <b>more</b> および <b>less</b> オプションで出力の詳細性を制御します。引数を指定しないでこのサブコマンドを入力した場合、このサブコマンドの現行設定値が出力されます。                                                                 |
| <b>delay</b> <i>MilliSeconds</i>                           | 認証を必要とする要求に組み込まれたタイム・スタンプに追加する時間間隔を指定します。このサブコマンドは、長時間遅延ネットワーク・パス上またはクロックが同期化されていないコンピューター間で信頼性のないサーバーを再構成することができます。引数を指定しないでこのサブコマンドを入力した場合、このサブコマンドの現行設定値が出力されます。                                  |
| <b>host</b> <i>HostName</i>                                | 照会の送信先のホストを指定します。 <i>HostName</i> は、ホスト名でも数値アドレスでもかまいません。引数を指定しないでこのサブコマンドを入力した場合、このサブコマンドの現行設定値が出力されます。                                                                                             |
| <b>hostnames</b> <b>yes</b>   <b>no</b>                    | ホスト名を出力するか ( <b>yes</b> ) または数値アドレスを出力するか ( <b>no</b> ) を指定します。 <b>-n</b> フラグが使用されない場合は、 <b>yes</b> にデフォルト設定します。引数を指定しないでこのサブコマンドを入力した場合、このサブコマンドの現行設定値が出力されます。                                      |
| <b>keyid</b> <i>Number</i>                                 | 構成要求の認証に使用するサーバー・キー番号を指定します。引数を指定しないでこのサブコマンドを入力した場合、このサブコマンドの現行設定値が出力されます。                                                                                                                          |
| <b>ntpversion</b> <b>1</b>   <b>2</b>   <b>3</b>           | パケットをポーリングするときに使用する NTP バージョン・インプリメンテーションを指定します。デフォルト値は 3 です。引数を指定しないでこのサブコマンドを入力した場合、このサブコマンドの現行設定値が表示されます。<br><b>注:</b> モード 6 の制御メッセージおよびモードは、NTP バージョン 1 にはありませんでした。                              |
| <b>passwd</b>                                              | 構成要求の認証に使用する NTP サーバー認証パスワードを入力するようプロンプトを指示します。                                                                                                                                                      |
| <b>quit</b>                                                | <b>ntpq</b> 照会プログラムを終了します。                                                                                                                                                                           |
| <b>raw</b>                                                 | リモート・サーバーから受信したすべての結果をフォーマットせずに表示します。非 ASCII 文字を出力可能なフォーマットに変換するだけです。                                                                                                                                |
| <b>rmvars</b> <i>Variable</i> [ = <i>Value</i> ] [ ,... ]  | 内部データ・リストから除去される変数およびそのオプション値を指定します。複数の変数を除去する場合、リストをコンマで区切り、しかもスペースが入らないようにしなければなりません。                                                                                                              |
| <b>timeout</b> <i>MilliSeconds</i>                         | サーバー照会に対する応答にタイムアウト期間を指定します。デフォルトは 5000 ミリ秒です。引数を指定しないでこのサブコマンドを入力した場合、このサブコマンドの現行設定値が出力されます。<br><b>注:</b> <b>ntpq</b> 照会プログラムは、タイムアウトの後にそれぞれの照会を一度ずつ再試行するので、タイムアウトの待ち時間の合計は、設定されたタイムアウト値の 2 倍になります。 |

## 制御メッセージ・サブコマンド

NTP サーバーが認識する各ピアには、16 ビットの整数のアソシエーション ID が割り当てられます。ピア変数を運ぶ NTP 制御メッセージは、そのアソシエーション ID を含めることにより、その値が対応するピア

アを識別しなければなりません。0 のアソシエーション ID は特別なものであり、その変数がシステム変数であり、その名前が別のネーム・スペースから取り出されたものであることを示します。

**ntpq** 制御メッセージ・サブコマンドは、サーバーに送信された 1 つまたは複数の NTP モード 6 のメッセージになって、あるフォーマットで戻されたデータを出力します。現在インプリメントされたサブコマンドのほとんどは、単一メッセージを送信し、単一応答を待ちます。現行例外のサブコマンドには、必要なデータを獲得するためのプログラム済みの一連のメッセージを送信する **peers** サブコマンド、およびアソシエーションの範囲上で繰り返す **mreadlist** および **mreadvar** サブコマンドがあります。

#### **associations**

照会中のサーバーのスペック内ピア用のアソシエーション ID およびピア状況のリストを獲得して表示します。リストは次の列に印刷されます。

- 最初の列には、内部使用のためにアソシエーションに 1 から番号を振る索引が入っています。
- 2 番目の列には、サーバーが戻す実際のアソシエーション ID が入っています。
- 3 番目の列には、ピアの状況ワードが入っています。
- 残りの列には、状況ワードからデコードしたデータが入っています。

**注: associations** サブコマンドによって戻されたデータは、**ntpq** 照会プログラム内に内部的にキャッシュされます。複雑なアソシエーション ID を使用するサーバーを扱うときは、**&index** というフォーマットにして、アソシエーション ID の代替として索引を引数のように使用します。

**clockvar** [ *AssocID* ] [ *Variable* [ =*Value* ], ... ]

または

**cv** [ *AssocID* ] [ *Variable* [ =*Value* ], ... ]

サーバーのクロック変数のリストを表示します。ラジオ・クロックまたは他の外部同期を持つサーバーは、これに対して肯定的に応答します。システム・クロック変数を要求するには、*AssocID* をブランクのままにするか、または 0 を指定します。サーバーがクロックを疑似ピアとして扱い、一度に複数のクロックを接続する可能性がある場合、該当のピア・アソシエーション ID を参照すると、特定のクロックの変数が表示されます。変数リストを省略すると、サーバーはデフォルトの変数表示を戻します。

#### **lassociations**

サーバーが状態を保持しているすべてのアソシエーション用のアソシエーション ID およびピア状況のリストを表示します。このサブコマンドは、スペック外のクライアント・アソシエーションのために状態を保存するサーバー用であるという点のみが **associations** サブコマンドと異なります。

#### **lpassociations**

内部的にキャッシュされたアソシエーションのリストから、スペック外のクライアント・アソシエーションを含む、すべてのアソシエーション用のデータを表示します。

#### **lpeers**

サーバーが **peers** サブコマンドの類似サブコマンド用に状態を保持するすべてのアソシエーションの要約を表示します。これは、スペック外のクライアント・サーバーからピアのより長いリストを作成することがあります。

**mreadvar** *AssocID* *AssocID* [ *Variable* [ =*Value* ], ... ]

または

**mr** *AssocID* *AssocID* [ *Variable* [ =*Value* ], ... ]

**mreadlist** *AssocID* *AssocID*

または

**mr** *AssocID* *AssocID*

与えられたゼロ以外のアソシエーション ID の範囲でそれぞれのサーバーに内部変数リスト内で指定されたピア変数の値を表示します。最新のアソシエーション・コマンドでキャッシュしたアソシエーション・リストが、その範囲を決定します。

#### **opeers**

**peers** サブコマンドの旧フォーマットです。参照 ID をローカル・インターフェース・アドレスに置き換えます。

## passociations

内部的にキャッシュされたアソシエーションのリストからスペック内ピアに関するアソシエーション・データを表示します。このサブコマンドは、新規の照会を作成するのではなく内部的に格納されたデータを表示するという点を除き、**associations** サブコマンドと同じように動作します。

## peers

サーバーのスペック内ピアのリストおよび各ピアの状態の要約を表示します。要約情報には、以下の情報が含まれます。

- リモート・ピアのアドレス
- 参照 ID (不明参照 ID の場合には 0.0.0.0)
- リモート・ピアの階層 (階層 16 は、リモート・ピアが同期化されていないことを示します)
- ピアのタイプ (ローカル、ユニキャスト、マルチキャスト、または ブロードキャスト)
- 最後にパケットを受信した時刻、ポーリング間隔 (秒数)
- ポーリング間隔 (秒)
- 到達可能レジスター (8 進数)
- ピアの現在予測される遅延、オフセット、および分散 (ミリ秒)。

左マージンの文字は、クロック選択プロセスにおけるこのピアの処理結果を下記のとおり示します。

### スペース

高層または正常性検査の失敗、あるいはその両方のために廃棄されました。

- x** 交差アルゴリズムによって疑似ティックが指定されました。
- .** 候補リストの終わりから選択されました。
- クラスタ化アルゴリズムによって廃棄されました。
- +** 最終選択セットに組み込まれました。
- #** 同期用に選択されましたが、最大距離を超えています。
- \*** 同期用に選択されました。
- o** **pps** シグナルを使用する同期用に選択されました。

ホスト・フィールドの内容は、ホスト名、IP アドレス、パラメーターまたは REFCLK(*ImplementationNumber* , *Parameter*) を指定した参照クロックのインプリメンテーション名であることがあります。IP アドレスが表示されるのは、**hostnames no** を使用したときのみです。

### 注:

**peers** サブコマンドは、受け取る応答の値を解析する能力に依存しています。データ・フォーマットを十分に制御しないサーバーとの動作に時々失敗することがあります。

**peers** サブコマンドは、非アトミックであるため、コマンドを発生し終了する無効なアソシエーションに関する疑似エラー・メッセージになることが時々あります。

## pstatus AssocID

読み取り状況要求を送信して、与えられたアソシエーションでサーバーのピア変数の名前および値を表示します。出力は、16 進数と英語の両方で変数に先行するヘッダーを表示します。

|                                                                                                                                                                  |                                                                                                                                             |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <b>readlist</b> [ <i>AssocID</i> ]<br>または<br><b>rl</b> [ <i>AssocID</i> ]                                                                                        | 与えられたアソシエーションでサーバーの内部変数リストのピア変数の値を表示します。システム変数を要求するには、 <i>AssocID</i> をブランクにしておくか、または 0 を入力します。内部変数リストが空の場合は、サーバーはデフォルトの変数表示を戻します。          |
| <b>readvar</b> [ <i>AssocID</i> ] [ <i>Variable</i> ] [ <i>=Value</i> ], ... ]<br>または<br><b>rv</b> [ <i>AssocID</i> ] [ <i>Variable</i> [ <i>=Value</i> ], ... ] | 読み取り変数要求を送信して、与えられたアソシエーションでサーバーの指定されたピア変数の値を表示します。システム変数を要求するには、 <i>AssocID</i> をブランクにしておくか、または 0 を入力します。変数リストを省略すると、サーバーはデフォルトの変数表示に戻します。 |
| <b>writevar</b> [ <i>AssocID</i> ] [ <i>Variable</i> ] [ <i>=Value</i> ], ... ]                                                                                  | 書き込み変数要求を送信して、与えられたアソシエーションでサーバーに指定されたピア変数の値を書き込みます。                                                                                        |
| <b>writelist</b> [ <i>AssocID</i> ]                                                                                                                              | 与えられたアソシエーションでサーバーの内部変数リストにピア変数の値を書き込みます。                                                                                                   |

## ファイル

`/usr/sbin/ntpq`                    `ntpq` コマンドが入っています。

## 関連情報

コマンド: `ntptime`、`ntpdate`、`ntptrace`、`xntpd`

デーモン: `xntpd`

---

## ntptrace コマンド

### 目的

Network Time Protocol (NTP) ホストのチェーンをマスター時間ソースに戻ってトレースします。

### 構文

```
ntptrace [ -d ] [ -n ] [ -v ] [ -r Retries] [ -t TimeOut] [ Server]
```

### 説明

**ntptrace** コマンドは、指定された NTP サーバーがその時刻をどこから取得するかを決定し、NTP サーバーのチェーンをそれらのマスター時刻ソースまでさかのぼって追跡します。例えば、`stratum 0` サーバーまでさかのぼって追跡します。

### フラグ

|                   |                                 |
|-------------------|---------------------------------|
| <b>-d</b>         | 出力のデバッグをオンにします。                 |
| <b>-n</b>         | ホスト名ではなくホストの IP アドレスを出力します。     |
| <b>-r Retries</b> | ホストごとの再送試行回数を指定します。デフォルトは 5 です。 |
| <b>-t TimeOut</b> | 再送タイムアウトを秒数で指定します。デフォルトは 2 秒です。 |
| <b>-v</b>         | 詳細モードを指定します。                    |

## パラメーター

*Server* サーバーを指定します。デフォルトはローカル・ホストです。

## 終了状況

このコマンドは、以下の終了値を戻します。

0 正常終了。  
>0 エラーが発生しました。

## セキュリティ

アクセス制御: このコマンドを実行するには、システム・グループの一部でなければなりません。

イベントの監査: N/A

## 例

ローカル・ホスト NTP サーバーがその時刻を取得するロケーションをトレースするには、下記のように入力します。

```
ntptrace
```

下記のような出力が表示されます。

```
localhost: stratum 4, offset 0.0019529, synch distance 0.144135  
server2.bozo.com: stratum 2, offset 0.0124263, synch distance 0.115784  
usndh.edu: stratum 1, offset 0.0019298, synch distance 0.011993, refid  
'WWVB'
```

それぞれの行のフィールドは下記の情報を示します。

1. ホストのレイヤー。
2. **ntptrace** コマンドで測定したとおりの、そのホストとローカル・ホストの間のオフセット時間 (ローカル・ホストの場合にゼロになるとは限らないのは、このためです)。
3. クロックの時間の品質の測度である、ホストの同期距離。
4. 参照クロックの ID。これは、stratum (階層)-1 サーバーのみに適用されます。

時間はすべて秒数で与えられます。

## ファイル

`/usr/sbin/ntptrace` **ntptrace** コマンドが入っています。

## 関連情報

コマンド: **ntpq**、**ntpdate**、**xntpdc**

デーモン: **xntpd**



---

## nulladm コマンド

### 目的

アクティブなアカウント・データ・ファイルを作成します。

### 構文

`/usr/sbin/acct/nulladm [ File ...]`

### 説明

**nulladm** コマンドは、*File* パラメーターで指定されたファイルを作成し、読み取り (r) および書き込み (w) 許可をファイル・オーナーに与え、またグループおよび読み取り (r) 許可をほかのユーザーに与え、さらにファイル・オーナーおよびグループが **adm** であるようにします。**nulladm** コマンドはいろいろなアカウント・シェル・プロシージャによって呼び出されます。管理権限のあるユーザーは、このコマンドを使用して、`/var/adm/wtmp` ファイルなどアクティブ・データ・ファイルをセットアップできます。

注: 分散環境ではノード間でアカウント・ファイルを共用できません。各ノードに、いろいろなアカウント・ファイルの独自のコピーがなくてはなりません。

### セキュリティ

アクセス制御: このコマンドは、**adm** グループのメンバーのみに実行 (x) アクセス権限を与えます。

### ファイル

|                                |                        |
|--------------------------------|------------------------|
| <code>/usr/sbin/acct</code>    | アカウント・コマンドが入っています。     |
| <code>/var/adm/acct/sum</code> | アカウント・データ・ファイルが入っています。 |

### 関連情報

**acctmerg** コマンド、**prdaily** コマンド。

オペレーティング・システムおよびデバイスのマネージのシステム・アカウントのセクション。

オペレーティング・システムおよびデバイスのマネージのアカウント・サブシステムの設定のセクション。

パフォーマンス・マネージメントのコマンドとサブルーチンのモニターとチューニングのセクション。

---

## number コマンド

### 目的

数値の記述フォーマットを表示します。

### 構文

`number`



## 説明

**number** コマンドは、入力された数の数値表現を、言葉で書いたフォーマットに変換します。変換できる最大数は、正確に 66 桁です。以下に例を示します。

```
12345678
twelve million.
three hundred forty five thousand.
six hundred seventy eight.
```

この例では、12345678 と入力したものがコンピューターによって twelve million three hundred forty five thousand six hundred seventy eight に変換されています。

**number** コマンドは数を求めるプロンプト表示を出しません。つまり、一度開始した後は入力を待つだけでプロンプト指示はしません。このプログラムを終了するには、割り込み (Ctrl-C) またはファイル終わり (Ctrl-D) のキー・シーケンスを押します。

## ファイル

**/usr/games** システム・ゲームが入っています。

## 関連情報

**arithmetic** コマンド、**back** コマンド、**bj** コマンド、**craps** コマンド、**fish** コマンド、**fortune** コマンド、**hangman** コマンド、**moo** コマンド、**quiz** コマンド、**ttt** コマンド、**turnoff** コマンド、**turnon** コマンド、**wump** コマンド。

---

## od コマンド

### 目的

ファイルを指定されたフォーマットで表示します。

### 構文

**Type-String** を使用してファイルを表示し、出力をフォーマットする

```
od [ -v ] [ -A AddressBase ] [ -N Count ] [ -j Skip ] [ -t TypeString ... ] [ File ... ]
```

フラグを使用してファイルを表示し、出力をフォーマットする

```
od [ -a ] [ -b ] [ -c ] [ -C ] [ -d ] [ -D ] [ -e ] [ -f ] [ -F ] [ -h ] [ -H ] [ -i ] [ -l ] [ -l ] [ -L ] [ -o ] [ -O ] [ -p ] [ -P ] [ -s ] [ -v ] [ -x ] [ -X ] [ -S [ N ] ] [ -w [ N ] ] [ File ] [ [ + ] Offset [ . | b | B ] [ + ] Label [ . | b | B ] ] [ File ... ]
```

### 説明

**od** コマンドは *File* パラメーターで指定されたファイルを指定されたフォーマットで表示します。 *File* パラメーターを指定しないと、**od** コマンドは標準入力を読みます。複数タイプは、複数の **-bcCDdFfOoSstvXx** オプションを使用することで指定できます。

最初の構文フォーマットの場合、出力フォーマットは **-t** フラグで指定します。フォーマットのタイプを指定しない場合は、**-t o2** がデフォルトとして使用されます。

2 番目の構文フォーマットの場合、出力フォーマットはフラグを組み合わせで指定します。 *Offset* パラメーターは、ファイル出力を開始するファイル内の位置を指定します。デフォルトでは、*Offset* パラメーターは 8 進のバイトとして解釈されます。 *.* (ドット) 接尾部を付加した場合、パラメーターは 10 進数として解釈されます。パラメーターの先頭が *x* または *0x* で始まる場合は、10 進数として解釈されます。 *b* 接尾部をパラメーターに付加した場合、512 バイトのブロックとして解釈されます。 *B* 接尾部をパラメーターに付加した場合、1024 バイトのブロックとして解釈されます。

*Label* パラメーターは最初に表示されたバイトの疑似アドレスとして解釈されます。使用する場合は、*Offset* パラメーターの後に ( ) (小括弧) で囲んで指定します。接尾部の意味は *Offset* パラメーターの場合と同じです。

**od** コマンドが標準入力を読む場合は、*Offset* パラメーターと *Label* パラメーターの前に + (プラス記号) を付けなくてはなりません。

**LANG** および **LC\_ALL** などの環境変数を設定すると、**od** コマンドの作用に影響します。詳しくは、オペレーティング・システムおよびデバイスのマネージのロケール環境変数についてのセクションを参照してください。

## フラグ

最初のフォーマットのフラグは以下のとおりです。

### **-A** *AddressBase*

入力オフセット・ベースを指定します。*AddressBase* 変数は以下の文字の 1 つを指定します。

- d** オフセット・ベースを 10 進数で書きます。
- o** オフセット・ベースを 8 進数で書きます。
- x** オフセット・ベースを 16 進数で書きます。
- n** オフセット・ベースを表示しません。

**-A n** が指定されていない場合は、出力行は書き込まれる次の行の、入力ファイル間の累積の入力オフセットによって処理されます。さらに、書き込まれた最後のバイトの後のバイトのオフセットは、すべての入力データが処理された後に表示されます。 **-A** アドレス基本オプションおよび [*offset\_string*] オペランドを指定しないと、入力オフセット・ベースは 8 進数で表示されます。

### **-j** *Skip*

出力を表示する前に *Skip* 変数によって指定されたバイト数をスキップします。複数のファイルを指定する場合、**od** コマンドは出力を表示する前に連結した入力ファイルの指定バイト数をスキップします。結合された入力が、少なくともスキップ・バイトの長さがない場合は、**od** コマンドは診断メッセージを標準エラーに書き込み、ゼロ以外の状況で終了します。

デフォルトでは、*Skip* 変数の値は 10 進数として解釈されます。先頭に *0x* または *0X* がある場合、オフセットは 16 進数として解釈され、先頭に *0* がある場合は、オフセットは 8 進数として解釈されます。 *Skip* 変数の値に文字 **b**、**k**、**m** を付加した場合、オフセットは *Skip* 変数の値 (バイト) にそれぞれ 512、1024、1024\*1024 を掛けた値となります。

### **-N** *Count*

*Count* 変数で指定した入力バイト数までをフォーマットします。デフォルトでは、*Count* 変数の値は 10 進数として解釈されます。先頭に *0x* または *0X* がある場合は、16 進数として解釈されます。先頭に *0* がある場合は、8 進数として解釈されます。表示されるアドレスのベースは、*Count* オプションの引数のベースで暗黙指定されません。

## **-t** *TypeString*

出力タイプを指定します。*TypeString* 変数は、データを書き出すときに使用するタイプを指定する文字列です。複数のタイプを同じ *TypeString* 変数に連結し、**-t** フラグを複数回指定できます。出力行はタイプ指定文字を指定した順序で、指定された各タイプに対して書き込まれます。*TypeString* 変数は以下の文字で構成することができます。

**a** バイトを指定された文字として表示します。これらの文字の対応する名前を使用して、0 から 01777 の範囲の少なくとも 7 ビットを持つバイトが書き込まれます。

**c** バイトを文字として表示します。**c** タイプ文字列によって変換されるバイト数は、**LC\_CTYPE** ローカル・カテゴリーによって決まります。印字可能なマルチバイト文字は、文字の最初のバイトに対応する領域に書き込まれます。2 文字のシーケンス **\*\*** は、文字内の残りのバイトそれぞれに対応する領域に、文字が継続することを示して書き込まれます。以下の非グラフィック文字が C 言語エスケープ・シーケンスとして使用されます。

```
\ Backslash
\a Alert
\b Backspace
\f Form-feed
\n New-line character
\0 Null
\r Carriage return
\t Tab
\v Vertical tab
```

**d** バイトを符号付き 10 進数として表示します。デフォルトでは、**od** コマンドは C 言語タイプ **int** の対応するバイト数を変換します。**d** タイプの文字列の後には、出力タイプの各インスタンスによって変換されるバイト数を指定する符号なしの 10 進整数を指定できます。

オプションの **C**、**I**、**L**、**S** 文字を **d** オプションに付加し、それぞれ **char**、**int**、**long**、**short** タイプの項目に変換を適用するように指定できます。

**f** バイトを浮動小数点数として表示します。デフォルトでは、**od** コマンドは C 言語のタイプ **double** の対応するバイト数を変換します。**f** タイプの文字列の後には、出力タイプの各インスタンスによって変換されるバイト数を指定する符号なしの 10 進整数を指定できます。

オプションの **F**、**D**、**L** 文字を **f** オプションに付加し、それぞれ **float**、**double**、**long double** タイプのエントリーに変換を適用するように指定できます。

**o** バイトを 8 進数として表示します。デフォルトでは、**od** コマンドは C 言語タイプ **int** の対応するバイト数を変換します。**o** タイプの文字列の後には、出力タイプの各インスタンスによって変換されるバイト数を指定する符号なしの 10 進整数を指定できます。

オプションの **C**、**I**、**L**、**S** 文字を **o** オプションに付加し、それぞれ **char**、**int**、**long**、**short** タイプの項目に変換を適用するように指定できます。

**u** バイトを符号なし 10 進数として表示します。デフォルトでは、**od** コマンドは C 言語タイプ **int** の対応するバイト数を変換します。**u** タイプの文字列の後には、出力タイプの各インスタンスによって変換されるバイト数を指定する符号なしの 10 進整数を指定できます。

オプションの **C**、**I**、**L**、**S** 文字を **u** オプションに付加し、それぞれ **char**、**int**、**long**、**short** タイプの項目に変換を適用するように指定できます。

**x** バイトを 16 進数として表示します。デフォルトでは、**od** コマンドは C 言語タイプ **int** の対応するバイト数を変換します。**x** タイプの文字列の後には、出力タイプの各インスタンスによって変換されるバイト数を指定する符号なしの 10 進整数を指定できます。

オプションの **C**、**I**、**L**、**S** 文字を **x** オプションに付加し、それぞれ **char**、**int**、**long**、**short** タイプの項目に変換を適用するように指定できます。

2 番目のフォーマットのフラグは以下のとおりです。

**-a** バイトを文字として、その ASCII 名とともに表示します。**-p** フラグも指定すると、偶数パリティを持つバイトが下線付きで表示されます。**-P** フラグを指定すると、奇数パリティを持つバイトが下線付きで表示されます。それ以外の場合、パリティは無視されます。

**-b** バイトを 8 進数として表示します。

**-c** バイトを ASCII 文字として表示します。以下の非グラフィック文字は C 言語エスケープ・シーケンスとして表示されます。

```
\ Backslash
\a Alert
\b Backspace
\f Form-feed
\n New-line character
\0 Null
\r Carriage return
\t Tab
\v Vertical tab
```

その他の文字は 3 桁の 8 進数として表示されます。

**-C** 拡張文字を標準印字可能な ASCII 文字として表示し (適切な文字によるエスケープ文字列使用)、マルチバイト文字を 16 進フォーマットで表示します。

**-d** 16 ビットのワードを符号なし 10 進数値として表示します。

**-D** long ワードを符号なし 10 進数値として表示します。

**-e** long ワードを倍精度の浮動小数点として表示します (**-F** フラグと同じ)。

**-f** long ワードを浮動小数点として表示します。

**-F** long ワードを倍精度の浮動小数点として表示します (**-e** フラグと同じ)。

**-h** 16 ビットのワードを符号なし 16 進数値として表示します。

**-H** long ワードを符号なし 16 進数値として表示します。

**-i** 16 ビットのワードを符号付き 10 進数として表示します。

**-I** (i の大文字) long ワードを符号付き 10 進数として表示します。

**-l** (L の小文字) long ワードを符号付き 10 進数として表示します。

**-L** long ワードを符号付き 10 進数として表示します。

注: フラグの **-I** (i の大文字)、**-l** (L の小文字)、**-L** は同一です。

**-o** 16 ビットのワードを符号なし 8 進数として表示します。

**-O** long ワードを符号なし 8 進数値として表示します。

**-p** **-a** 変換時の偶数パリティを示します。

**-P** **-a** 変換時の奇数パリティを示します。

- s** 16 ビットのワードを符号付き 10 進数として表示します。
- S[N]** null バイトで終わる文字列を検索します。 *N* 変数は認識される最小の長さの文字列を指定します。 *N* 変数を省略すると、最小の長さは 3 文字となります。

**-v** フラグは両方のフォーマットで同じ意味を持ちます。

- v** すべての入力データを書きます。 デフォルトでは、直前の出力行と同じ出力行は出力されず、\* (アスタリスク) 1 つだけを含む行に置き換えられます。 **-v** フラグを指定すると、すべての行が出力されます。
- w [N]** 解釈して、出力行ごとに表示する入力バイト数を指定します。 **-w** フラグを指定しないと、表示行ごとに 16 バイトが読み取られます。 **-w** フラグを *N* 変数なしで指定すると、表示行ごとに 32 バイトが読み取られます。最大入力値は 4096 バイトです。4096 バイトよりも大きい入力値は最大値に再割り当てされます。
- x** 16 ビットのワードを 16 進数として表示します。
- X** long ワードを符号なし 16 進数値として表示します。 (**-H** フラグと同じ)

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0** すべての入力ファイルが正常に出力されました。
- >0** エラーが発生しました。

## 例

1. ファイルを 8 進数で一度に 1 ページずつ表示するには、以下のように入力します。

```
od a.out | pg
```

このコマンドは `a.out` ファイルを 8 進数フォーマットで表示し、`pg` コマンドを介して出力を送ります。

2. ファイルを一度に複数のフォーマットに変換するには、以下のように入力します。

```
od -t cx a.out > a.xcd
```

このコマンドは `a.out` ファイルの内容を 16 進フォーマット (**x**) および文字フォーマット (**c**) で `a.xcd` ファイルに書き込みます。

3. ファイルの途中から表示を開始するには (最初の構文フォーマットを使用)、以下のように入力します。

```
od -t acx -j 100 a.out
```

このコマンドは、`a.out` ファイルを、指定された文字 (**a**) フォーマット、文字 (**c**) フォーマット、および 16 進 (**x**) フォーマットで、100 バイト目から表示します。

4. ファイルの途中から表示を開始するには (2 番目の構文フォーマットを使用)、以下のように入力します。

```
od -bcx a.out +100.
```

このコマンドは、`a.out` ファイルを 8 進バイト (**-b**) フォーマット、文字 (**-c**) フォーマット、および 16 進 (**-x**) フォーマットで、100 バイト目から表示します。 オフセットの後の `.` (ピリオド) により 10 進数になります。ピリオドを指定しない場合は、出力は 64 (8 進数 100) バイト目から始まります。

## ファイル

`/usr/bin/od` `od` コマンドが入っています。

## 関連情報

`dbx` コマンド、`pg` コマンド。

*AIX 5L* バージョン 5.3 ナショナル・ランゲージ・サポート ガイドおよびリファレンスのナショナル・ランゲージ・サポートの概要のセクションおよびロケール環境変数についてのセクション。

---

## odmadd コマンド

### 目的

作成されたオブジェクト・クラスにオブジェクトを追加します。

### 構文

```
odmadd [ InputFile ... ]
```

### 説明

`odmadd` コマンドは 1 つまたは複数の *InputFile* ファイルを入力とし、スタンザ・ファイルで検索されたデータとともにオブジェクトをオブジェクト・クラスに追加します。各 *InputFile* ファイルは、オブジェクト・クラスに追加するオブジェクトを記述するデータを含む ASCII ファイルです。ファイルを指定しないと、入力は `stdin` (標準入力) から取ります。

追加先のクラスは ASCII 入力ファイルで指定されます。このファイルの一般フォーマットは以下のとおりです。

```
classname:
    descriptor1name = descriptor1value
    descriptor2name = descriptor2value
    descriptor3name = descriptor3value
```

```
class2name:
    descriptor4name = descriptor4value
```

```
.
.
.
```

入力ファイルには \ (バックスラッシュ) を入れることができます。これは C 言語の場合と同様に処理されます。入力ファイル内の文字列値とメソッド値は必ず " " (二重引用符) で囲みます。ディスクリプター値は複数行にまたがって指定できます。

### 例

`odmadd` コマンドが使用する ASCII 入力ファイルは以下のようになります。

```
Fictional_Characters:
Story_Star      = "Cinderella"
Birthday       = "Once upon a time"
Age            = 19
Friends_of     = Cinderella
Enemies_of     = "Cinderella"
```

```
Friend_Table:
Friend_of      = "Cinderella"
Friend        = "Fairy godmother"
```

```
Friend_Table:
Friend_of      = "Cinderella"
Friend        = "Mice"
```

```
Enemy_Table:
Enemy_of       = "Cinderella"
Enemy         = "Wicked sisters"
```

```
Enemy_Table:
Enemy_of       = "Cinderella"
Enemy         = "Mean stepmother"
```

先頭のファイルを `NewObjects` と命名すると、次のコマンドは既存のオブジェクト・クラスにこのオブジェクトを追加します。

```
odmadd NewObjects
```

**odmadd** コマンドで入力可能な ASCII ファイルの別の例は、「*AIX 5L* バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM のコードおよび出力例のセクションを参照してください。

## 関連情報

「*AIX 5L* バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」のプログラマーのためのオブジェクト・データ・マネージャー (ODM) の概要のセクション。

**odm\_add\_obj** サブルーチン。

「*AIX 5L* バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM コマンドとサブルーチンのリストのセクション。

「*AIX 5L* バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM のオブジェクト・クラスおよびオブジェクトについてのセクション。

「*AIX 5L* バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」のオブジェクト・クラスの作成方法のセクション。

「*AIX 5L* バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」のオブジェクト・クラスへのオブジェクトの追加方法のセクション。

「*AIX 5L* バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」のオブジェクト・クラスおよびオブジェクトの保管方法のセクション。

---

## odmchange コマンド

### 目的

指定されたオブジェクト・クラスで選択されたオブジェクトの内容を変更します。

### 構文

```
odmchange -o ObjectClass [ -q Criteria] [ InputFile]
```



## 説明

修正対象となるオブジェクト・クラス、検索基準、および新しいオブジェクト (変更が必要な属性についてのみ) を指定すると、**odmchange** コマンドは検索基準を満たすすべてのオブジェクトを修正します。

*InputFile* ファイルは、**odmadd** コマンドの場合の *InputFile* ファイル (ASCII 入力ファイル) と同じフォーマットになっています。

## フラグ

- o** *ObjectClass*                    修正するオブジェクト・クラスを指定します。
- q** *Criteria*                      オブジェクト・クラスからオブジェクトを選択するために使用する基準を指定します。修飾基準については、「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM オブジェクト検索についてのセクションを参照してください。基準を指定しないと (**-q** フラグを指定しないと)、オブジェクト・クラス内のすべてのオブジェクト・エントリーが変更されます。

## 関連情報

AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグのプログラマーのためのオブジェクト・データ・マネージャー (ODM) の概要のセクション。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」のプログラマーのためのオブジェクト・データ・マネージャー (ODM) の概要のセクション。

**odmadd** コマンド。

**odm\_change\_obj** サブルーチン。

AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグの ODM ディスクリプターについてのセクション。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM コマンドとサブルーチンのリストのセクション。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM のコードおよび出力例のセクション。

---

## odmcreate コマンド

### 目的

ODM アプリケーション開発に必要な **.c** ファイル (ソース・ファイル) と **.h** ファイル (組み込みファイル) を作成し、空のオブジェクト・クラスを作成します。

### 構文

**odmcreate** [ **-p** ] [ **-c** | **-h** ] *ClassDescriptionFile*

## 説明

**odmcreate** コマンドは ODM クラス・コンパイラーです。このコマンドは、ユーザーが特定のアプリケーションで使用したいオブジェクトを記述する ASCII ファイルを入力として使います。**odmcreate** コマンドは実行の一部として空のオブジェクト・クラスを作成できます。

**odmcreate** コマンドの出力は **.h** ファイル (組み込みファイル) です。このファイルには、ASCII の *ClassDescriptionFile* ファイルで定義されたオブジェクト・クラスに関する C 言語定義が入っています。生成された組み込みファイルは、アプリケーションが ODM に保管されたオブジェクトにアクセスするときに使います。**odmcreate** コマンドは **.c** ファイルも生成します。このファイルはコンパイルして、アプリケーションでバインドしなくてはなりません。**.c** ファイルには、ODM が実行時に内部で使う構造体と定義が入っています。

*ClassDescriptionFile* パラメーターは 1 つまたは複数のオブジェクト・クラスの記述を含む ASCII ファイルを指定します。*ClassDescriptionFile* パラメーターの一般的な構文は以下のとおりです。

```
file           : classes
classes        : class | classes class
class          : head body tail
head           : struct ClassName {
tail           : }
body           : elements
elements       : elements | elements element
element        : char DescriptorName [ DescriptorSize ];

               vchar DescriptorName [ DescriptorSize ];

               binary DescriptorName [ DescriptorSize ];

               short DescriptorName ;

               long DescriptorName ;

               long64 または int64 または ODM_LONG_LONG DescriptorName ;

               method DescriptorName ;

               link StdClassName StdClassName ColName DescriptorName ;
```

*ClassDescriptionFile* ファイルのデフォルトの接尾部は **.cre** です。**odmcreate** コマンドに接尾部を指定しないと、**.cre** 接尾部が付加されます。このファイルでは、**-p** フラグを指定して実行すると、C 言語のコメントを入れることができます。**-p** フラグを指定してこのファイルで C 言語プリプロセッサを実行すると、プリプロセスできる **#define** 行および **#include** 行を取り込むことができます。

**注:** ODM データベースは 32 ビット・データベースです。long タイプは、クラス記述ファイルで使われる場合 32 ビット・データ・エントリーです。long64 または int64 タイプは、クラス記述ファイルで使われる場合 64 ビット・データ・エントリーです。生成されたファイルは 32 ビットと 64 ビットのアプリケーションの両方で同様に機能します。

## フラグ

- c** 空のオブジェクト・クラスだけを作成します。C 言語の **.h** ファイルと **.c** ファイルは生成しません。
- h** **.c** ファイルと **.h** ファイルだけを生成します。空のクラスは作成しません。
- p** *ClassDescriptionFile* ファイルで C 言語プリプロセッサを実行します。

## 例

FileName.cre という *ClassDescriptionFile* ファイルがあると仮定して、次のコマンドはオブジェクト・クラスを作成します。

```
odmcreate FileName.cre
```

以下に示すのは、FileName.cre ソース・ファイルと、結果として生成される .h ファイルです。

```
/* This is an example odmcreate input file */
/* FileName.cre */

    class Class2 {
        char keys[32];
        method card;
        long cash;
    };
class TstObj {
    long a;
    char b[80];
    link Class2 Class2 card Class2Ln;
};

/* End of FileName.cre */

/* This is the generated header file FileName.h */
#include <odmi.h>

struct Class2 {
    long _id;          /* unique object id within object class */
    long _reserved;   /* reserved field */
    long _scratch;    /* extra field for application use */
    char keys[32];
    char card[256];   /* method */
    long cash;
};
#define Class2_Descs 3

extern struct Class Class2_CLASS[];
#define get_Class2_list (a,b,c,d,e) (struct Class2 * ) odm_get_list (a,b,c,d,e)

struct TstObj {
    long _id;          /* unique object id within object class */
    long _reserved;   /* reserved field */
    long _scratch;    /* extra field for application use */
    long a;
    char b[80];
    struct Class2 *Class2Ln; /* link */
    struct objlistinfo *Class2Ln_info; /* link */
    char Class2Ln_Lvalue[256]; /* link */
};
#define TstObj_Descs 3

extern struct Class TstObj_CLASS[];
#define get_TstObj_list (a,b,c,d,e) (struct TstObj * ) odm_get_list (a,b,c,d,e)

/* End of generated header file FileName.h */
```

*ClassDescriptionFile* パラメーターと、結果として生成される .h ファイルの別の例は、「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM のコードおよび出力例のセクションを参照してください。

## 関連情報

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」のオブジェクト・データ・マネージャー (ODM) の概要のセクション。

**odm\_create\_class** サブルーチン。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM コマンドとサブルーチンのリストのセクション。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM のオブジェクト・クラスおよびオブジェクトについてのセクション。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM ディスクリプターについてのセクション。

AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグのオブジェクト・クラスの作成方法のセクション。

AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグのオブジェクト・クラスへのオブジェクトの追加方法のセクション。

AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグのオブジェクト・クラスおよびオブジェクトの保管方法のセクション。

---

## odmdelete コマンド

### 目的

指定されたオブジェクト・クラスから選択したオブジェクトを削除します。

### 構文

```
odmdelete -o ObjectClass [ -q Criteria]
```

### 説明

削除元となるオブジェクト・クラスと検索基準を指定すると、**odmdelete** コマンドはその検索基準を満たすすべてのオブジェクトを削除します。

### フラグ

**-o** *ObjectClass*

削除元となるオブジェクト・クラスを指定します。

**-q** *Criteria*

オブジェクト・クラスからオブジェクトを選択するために使用する基準を指定します。修飾基準については、「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM オブジェクト検索についてのセクションを参照してください。基準を指定しないと (**-q** フラグを指定しないと)、すべてのオブジェクトが削除されます。

## 関連情報

AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグのプログラマーのためのオブジェクト・データ・マネージャー (ODM) の概要のセクション。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」のオブジェクト・データ・マネージャー (ODM) の概要のセクション。

**odm\_rm\_obj** サブルーチン。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM のオブジェクト・クラスおよびオブジェクトについてのセクション。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM コマンドとサブルーチンのリストのセクション。

---

## odmdrop コマンド

### 目的

オブジェクト・クラスを除去します。

### 構文

**odmdrop -o** *ClassName*

### 説明

**odmdrop** コマンドは、オブジェクト・クラス全体とそのすべてのオブジェクトを削除します。削除するクラスにほかのオブジェクト・クラスがリンクされているかどうかは検査しません。

### フラグ

**-o** *ClassName*                      削除するオブジェクト・クラスを指定します。

### 例

MyObjectClass というオブジェクト・クラスがあると仮定した場合、次のコマンドはこのオブジェクト・クラスを削除します。

```
odmdrop -o MyObjectClass
```

## 関連情報

AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグのプログラマーのためのオブジェクト・データ・マネージャー (ODM) の概要のセクション。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」のオブジェクト・データ・マネージャー (ODM) の概要のセクション。

**odm\_rm\_class** サブルーチン。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM のオブジェクト・クラスおよびオブジェクトについてのセクション。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM コマンドとサブルーチンのリストのセクション。

---

## odmget コマンド

### 目的

指定されたオブジェクト・クラスからオブジェクトを検索して **odmadd** の入力ファイルに入れます。

### 構文

```
odmget [ -q Criteria] ObjectClass...
```

### 説明

**odmget** コマンドは、検索基準とオブジェクト・クラスのリストを入力として使い、指定されたオブジェクト・クラスから選択されたオブジェクトを取り出し、ASCII **odmadd** 入力ファイルを標準出力に書きます。

### フラグ

**-q Criteria** 1 つまたは複数のオブジェクト・クラスからオブジェクトを選択するために使用する検索基準を指定します。検索基準については、「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM オブジェクト検索についてのセクションを参照してください。基準を指定しないと (**-q** フラグを指定しないと)、オブジェクト・クラス内のすべてのオブジェクトが取り出されます。

### 例

次の **odmget** コマンドは、文字列 `Fairy Godmother` と等しい Others ディスクリプターのある既存のオブジェクト・クラス Supporting\_Cast\_Ratings からオブジェクトを取り出します。

```
odmget -q"Others='Fairy Godmother'" Supporting_Cast_Ratings
```

ASCII **odmadd** 入力ファイルの例は、**odmadd** コマンドのセクションまたは「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM のコードおよび出力例のセクションを参照してください。

### 関連情報

**odmadd** コマンド。

AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグのプログラマーのためのオブジェクト・データ・マネージャー (ODM) の概要のセクション。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM のオブジェクト・クラスおよびオブジェクトについてのセクション。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM のコードおよび出力例のセクション。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM コマンドとサブルーチンのリストのセクション。

---

## odmshow コマンド

### 目的

オブジェクト・クラスの定義を画面に表示します。

### 構文

**odmshow** *ObjectClass*

### 説明

**odmshow** コマンドは、オブジェクト・クラス名 (*ObjectClass*) を入力として使い、クラス記述を画面に表示します。クラス記述のフォーマットは、**odmcreate** コマンドへの入力として使われるフォーマットです。

### 例

MyObjectClass というオブジェクト・クラスがあると仮定した場合、次のコマンドは MyObjectClass の記述を画面に表示します。

```
odmshow MyObjectClass
```

また、出力リストの例は、**odmcreate** コマンドのセクションまたは「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM のコードおよび出力例のセクションを参照してください。

### 関連情報

AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグのプログラマーのためのオブジェクト・データ・マネージャー (ODM) の概要のセクション。

**odmcreate** コマンド。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM のオブジェクト・クラスおよびオブジェクトについてのセクション。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM のコードおよび出力例のセクション。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の ODM コマンドとサブルーチンのリストのセクション。

---

## on コマンド

### 目的

リモート・システム上でコマンドを実行します。



## 構文

`/usr/bin/on [ -i ] [ -d ] [ -n ] Host Command [ Argument ... ]`

## 説明

**on** コマンドは、プログラムを実行する環境と類似する環境にあるほかのシステム上でコマンドを実行します。**on** コマンドはローカル環境変数をリモート・コンピューターに渡し、それによって現在の作業ディレクトリーを保持します。**on** コマンドを使用するときは、両方のユーザーが同じユーザー識別を持たなくてはなりません。相対パス名は現行ファイルシステム内でのみ有効です。コマンドは一方のコンピューターから発行されて、他方のコンピューター上で実行されるので、絶対パス名を使用すると問題が発生することがあります。

標準入力、リモート・コマンドの標準入力に接続されます。リモート・コマンドからの標準出力と標準エラーは、**on** コマンドの対応するファイルに送られます。`root` ユーザーは **on** コマンドを実行できません。

**重要:** 作業ディレクトリーがネットワーク・ファイルシステム (NFS) 上にリモート・マウントされている場合、キー順 `Ctrl-Z` を使用するとウィンドウが停止します。

## フラグ

- d** デバッグ・モードを指定します。作業の進行中に状況メッセージを表示します。
- i** 対話モードを指定します。リモート・エコーおよび特殊文字処理を使用します。このオプションは、端末と対話するプログラムに必要です。すべての端末モードとウィンドウ・サイズの変更は増加されます。
- n** 入力を指定しません。このオプションにより、リモート・プログラムは、標準入力から読む場合ファイル終わり (EOF) メッセージを受け取ります。このフラグは、コマンドをジョブ制御によりバックグラウンドで実行するときに必要です。

## 例

**ls -al** コマンドをほかのコンピューターで実行し、処理中の状況メッセージを端末に表示するには、以下のように入力します。

```
on -d zorro ls -al
```

この例では、**on** コマンドは **ls** コマンドを `zorro` というワークステーションで実行します。

## ファイル

`/etc/exports`

サーバーがエクスポートできるディレクトリーをリストします。

`/etc/inetd.conf`

**inetd** デーモンがインターネット・サービス要求を処理する方法を定義します。

## 関連情報

**rex**d デーモン。

ネットワークおよびコミュニケーションのマネージのネットワーク・ファイルシステム (NFS) の概要のセクション。

ネットワークおよびコミュニケーションのマネージ の NFS コマンドのリストのセクション。

---

## OS\_install コマンド

### 目的

**OS\_install** オブジェクトについてネットワーク・インストール操作を行います。

### 構文

伝統的な使用法:

```
OS_install { -o Operation } [ -a attr=value... ] {ObjectName}
```

システム・プラン・インストール用 (System Plan モード):

```
OS_install -i sysplan { -x sysplan.xml } [ -d ]
```

**OS\_install** オブジェクトのリスト用 (List モード):

```
OS_install -l [ -v ] [ -t object_type | object_name ]
```

### 説明

**OS\_install** コマンドは、**OS\_install** オブジェクトについてネットワーク・インストール操作を実行します。操作のタイプは、*ObjectName* パラメーターで指定するオブジェクトのタイプによって異なります。*ObjectName* パラメーターによりポイントされるオブジェクトは、以下の 3 つのタイプのいずれかです。すなわち、クライアント、**OS\_Resource**、または **ControlHost** のいずれかです。コマンド操作には、クライアント・マシンにオペレーション・システムをネットワーク・インストールするための、**OS\_install** オブジェクトの作成と管理が含まれます。

**OS\_install** は、操作を指定するのではなく、**-i sysplan** フラグを引き渡すことにより、System Plan モードで実行することができます。これにより、複数の **OS\_install** 操作を結合して 1 つの XML 文書にすることが可能となります。

**OS\_install** の List モードは、**OS\_install** 環境におけるオブジェクトの現在の構成をリストする場合に使用します。

### フラグ

**-a** *attr=value*

指定された属性に指定された値を割り当てます。

**Operations** は、特定の操作のための必須ならびにオプションの属性をリストします。

**-d**

System Plan モードの間に作成されたすべての **OS\_install** オブジェクトを、すべての操作の完了後に破壊します。

**-i sysplan**

System Plan モードを指定します。

**-l**

デフォルトで、その環境におけるすべての **OS\_install** オブジェクトをリストします。

**-o** *Operation*

**OS\_install** オブジェクトに対して行う操作を指定します。

-t *object\_type* | *object\_name*

-v

-x *sysplan.xml*

-I フラグにより戻されるリストを、*object\_name* で指定された *object\_type* タイプのオブジェクトのみ、あるいは単一の **OS\_install** オブジェクトに制限します。

-I フラグにより戻されるリストを表示します。  
システム・プランを含んだ XML ファイルを指定します。

## 操作

| 操作                                                                             | 説明                      | 必須属性                                                                                                                                                                              | オプションの属性                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------------------------------------------------------------------|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>define_client</b> [-a <i>attr=value...</i> ]<br>{ <i>ClientObjectName</i> } | 新しいクライアント・オブジェクトを定義します。 | <b>ip_addr</b><br>クライアントの IP アドレス。<br><b>mac_addr</b><br>クライアントのネットワーク・インターフェースの MAC アドレス。<br><b>gateway</b><br>クライアントの IP ゲートウェイ。<br><b>subnet_mask</b><br>クライアントの IP サブネット・マスク。 | <b>adapter_speed</b><br>クライアントのネットワーク・アダプターのスピード。<br><b>adapter_duplex</b><br>クライアントのネットワーク・アダプターの二重設定。<br><b>lpar</b><br>クライアントをインストールする LPAR 名。<br><b>profile</b><br>クライアントのために使用する LPAR プロファイル。<br><b>managed_system</b><br>LPAR を含んだ管理下のシステムの名前。<br><b>disk_location</b><br>クライアントをインストールするためのディスクの場所。<br><b>ctrl_host</b><br>当クライアントのためのハードウェア制御ホスト (Hardware Control Host) オブジェクトの名前。 |

| 操作                                                                                     | 説明                                                                                                                                                   | 必須属性                                                                                                           | オプションの属性                                                                                                                        |
|----------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| <b>define_resource</b> [-a <i>attr=value...</i> ]<br>{ <i>ResourceObjectName</i> }     | 新しい <b>OS_Resource</b> オブジェクトを定義します。                                                                                                                 | <b>type</b> Linux の AIX。<br><b>version</b> OS バージョン。                                                           | <b>location</b><br><b>OS_Resource</b> を常駐させようとしている絶対パス。<br><b>source</b> インストール・イメージのソース。<br><b>configfile</b><br>インストール構成ファイル。 |
| <b>define_ctrl_host</b> [-a <i>attr=value...</i> ]<br>{ <i>ControlHostObjectName</i> } | 新しいハードウェア制御ホスト・オブジェクトを定義します。                                                                                                                         | <b>communication_method</b><br>ssh、rsh、または local。<br><b>hostname</b><br>制御ホストのホスト名<br><b>type</b> hmc または ivm。 | なし。                                                                                                                             |
| <b>allocate</b> [-a <i>attr=value...</i> ]<br>{ <i>ClientObjectName</i> }              | <b>OS_Resource</b> をクライアント・オブジェクトに割り当てます。両方のオブジェクトが <b>OS_install</b> 環境に既に存在していなければなりません。クライアント・オブジェクトに <b>OS_Resource</b> が既に割り当てられていると、エラーが発生します。 | <b>os_resource</b><br>クライアント・オブジェクトに割り当てる既存の <b>OS_Resource</b> オブジェクト。                                        | なし。                                                                                                                             |
| <b>netboot</b> { <i>ClientObjectName</i> }                                             | クライアント・オブジェクトのハードウェア制御ホストに対して、ネットワーク・ブートを開始するよう指示します。                                                                                                | なし。                                                                                                            | なし。                                                                                                                             |
| <b>monitor_installation</b><br>{ <i>ClientObjectName</i> }                             | クライアント・オブジェクトのインストール状況をモニターします。                                                                                                                      | なし。                                                                                                            | なし。                                                                                                                             |
| <b>deallocate</b><br>{ <i>ClientObjectName</i> }                                       | 割り当て操作によりクライアント・オブジェクトに割り当てられた <b>OS_Resource</b> を割り当て解除します。                                                                                        | なし。                                                                                                            | なし。                                                                                                                             |
| <b>remove</b> { <i>ObjectName</i> }                                                    | オブジェクトを <b>OS_install</b> 環境から除去します。                                                                                                                 | なし。                                                                                                            | なし。                                                                                                                             |

## 終了状況

0

コマンドは正常に実行されました。

>0

エラーが発生しました。

## 例

1. クライアント・オブジェクトを定義するには、以下のようなコマンドを入力します。

```
OS_install -o define_client -a ip_addr=128.0.64.117 -a mac_addr=ab:cc:de:10:23:45 -a \
gateway=128.0.64.1 -a subnet_mask=255.255.255.0 -a ctrl_host=myhmc -a lpar=AIX1 -a \
profile=AIX1 -a managed_system=myMngSys myclient01
```

上述のクライアント・オブジェクトは、管理下のシステム内の論理区画です。

2. **OS\_Resource** オブジェクトを定義するには、以下のようなコマンドを入力します。

```
OS_install -o define_resource -a location=/images/AIX/53ML3 -a type=AIX -a version=53ML3 my53resource
```

3. 上述の例で定義した **OS\_Resource** オブジェクトをクライアント・オブジェクトに割り当てるには、以下のようなコマンドを入力します。

```
OS_install -o allocate -a os_resource=my53resource myclient01
```

4. 上述の例で割り当てられた **my53resource** クライアント・オブジェクトを割り当て解除するには、以下のように入力します。

```
OS_install -o deallocate myclient01
```

5. **ctrl\_host** 属性に対して指定する **ControlHost** オブジェクトを定義するには、以下のようなコマンドを入力します。

```
OS_install -o define_ctrl_host -a type=hmc -a hostname=hmc_hostname -a communication_method=ssh myhmc
```

上述の例は、最初の例の **ctrl\_host** 属性と同じ名前を共有していますが、**define\_client** 操作は、未定義の **ControlHost** オブジェクトを **ctrl\_host** 属性に対して指定することを許します。この場合、**ControlHost** オブジェクトは、そのクライアントに対する **netboot** 操作が呼び出される前に定義されていなければなりません。**ControlHost** オブジェクトを定義するには、以下のように入力します。

```
OS_install -o netboot myclient01
```

6. **myclient01** インストール済み環境を表示するには、以下のように入力します。

```
OS_install -o monitor_installation myclient01
```

7. **my53resource** オブジェクトの定義を除去するには、次のように入力します。

```
OS_install -o remove my53resource
```

8. **myclient01** オブジェクトの定義を除去するには、次のように入力します。

```
OS_install -o remove myclient01
```

**OS\_Resource** オブジェクトが指定されている場合は、このオブジェクトの **location** 属性で指定されたファイルシステム・ディレクトリに存在するすべての **OS** イメージも **remove** 操作により除去されます。

## 位置

**/usr/sbin/OS\_install**

## ファイル

**/var/osinstall**

**OS\_install** 環境のための構成ファイルを含んだディレクトリ。

## 関連情報

インストールおよび移行の Installing with Network Installation Management のセクション。

---

## oslevel コマンド

### 目的

このシステムの最新のインストール・レベル (テクノロジー・レベル、保守レベル、および Service Pack) をレポートします。

### 構文

```
oslevel [ -l Level | -g Level | -q ] [-r | -s ] [-f]
```

### 説明

**oslevel** コマンドは、システムにインストールされたすべてのファイルセットのサブセットを使用して、オペレーティング・システムのテクノロジー・レベルと Service Pack を報告します。これらのファイルセットには、基本オペレーティング・システム (BOS)、基本デバイス、基本プリンター、および X11 があります。

**oslevel** コマンドはテクノロジー・レベルについての情報も表示します。例えば、指定されたテクノロジー・レベルまたは Service Pack がないファイルセットはどれかといった情報です。

### フラグ

|                        |                                                                                                                                   |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <b>-l</b> <i>Level</i> | <i>Level</i> パラメーターで指定したテクノロジー・レベルまたは Service Pack よりも以前のファイルセットをリストします。                                                          |
| <b>-f</b>              | <b>oslevel</b> コマンドがこの操作のためのキャッシュを強制的に再作成するようにします。                                                                                |
| <b>-g</b> <i>Level</i> | <i>Level</i> パラメーターで指定したテクノロジー・レベルまたは Service Pack より以降のファイルセットをリストします。                                                           |
| <b>-q</b>              | 既知のテクノロジー・レベル ( <b>-r</b> フラグを使用時) または Service Pack ( <b>-s</b> フラグを使用時) の名前をリストします。これらは、 <b>-l</b> または <b>-g</b> フラグを使用して指定できます。 |
| <b>-r</b>              | すべてのフラグをテクノロジー・レベルに適用します。                                                                                                         |
| <b>-s</b>              | すべてのフラグを Service Pack に適用します。                                                                                                     |

フラグが指定されていない場合は、基本システム・ソフトウェアは、完全に、**oslevel** コマンドの出力でリストされたレベルまたはそれより上のレベルです。

### 例

1. システムの基本レベルを判別するには、以下のように入力します。

```
oslevel
```

出力は次のようになります。

```
5.3.0.0
```

2. このシステムの現行バージョンの AIX に対して適用可能な最高のテクノロジー・レベルを判別するには、次のように入力します。

```
oslevel -r
```

3. システム上の、知られているすべてのテクノロジー・レベルをリストするには、次のように入力します。

```
oslevel -rq
```

戻されるレベルは、[ **-r -l** ] または [ **-r -g** ] フラグを指定して使用可能であり、以下と類似していません。

```
Known Recommended Maintenance Levels
```

```
-----
```

```
5300-02
```

```
5300-01
```

```
5300-00
```

4. AIX バージョン 5.3 テクノロジー・レベル 1 よりも低いソフトウェアをリストするには、次のように入力します。

```
oslevel -r -l 5300-01
```

5. AIX バージョン 5.3 テクノロジー・レベル 1 よりも高いソフトウェアをリストするには、次のように入力します。

```
oslevel -r -g 5300-01
```

6. システムの現行テクノロジー・レベルに対して、到達している最高の Service Pack を判別するには、次のように入力します。

```
oslevel -s
```

7. システム上の既知の Service Pack をリストするには、次のように入力します。

```
oslevel -sq
```

戻されるそのレベルは、[ **-s -l** ] または [ **-s -g** ] フラグを指定して使用可能であり、以下と類似しています。

```
Known Service Packs
```

```
-----
```

```
5300-06-02-0727
```

```
5300-06-01-0722
```

```
5300-06-00-0000
```

## ファイル

`/usr/bin/oslevel` `oslevel` コマンドが入っています。

## 関連情報

`lspp` コマンド。

---

## ospf\_monitor コマンド

### 目的

OSPF ゲートウェイをモニターします。

### 構文

```
ospf_monitor mon_db_file
```

### 説明

**ospf\_monitor** コマンドは、OSPF ルーターを照会するのに使用します。**ospf\_monitor** コマンドは、対話モードで動作します。このコマンドを使用すれば、ユーザーは、さまざまな OSPF ルーターを照会し、



入出力統計情報、エラー・ログ、リンク状態データベース、AS 外部データベース、OSPF 経路指定テーブル、構成済み OSPF インターフェース、および OSPF 隣接局に関する詳細情報を提供することが可能になります。

**ospf\_monitor** リモート・コマンドの宛先を構成しているレコードからなるデータベースの完全なパス名を *mon\_db\_file* で指定します。それぞれの宛先レコードは単一行入力域で、宛先 IP アドレス、宛先ホスト名、および OSPF 認証キー（認証が宛先によって活動化されている場合）がここにリストされます。認証キーが宛先レコード内に収められている場合があるため、このデータベースへの全般的なアクセスを制限するようお勧めします。

OSPF データベースおよびパケット・フォーマットの詳細は、RFC-1583「OSPF 仕様、バージョン 2」を参照してください。

## コマンド

対話モードに入ったら、**ospf\_monitor** は、'[ # ] dest command params >' プロンプトを表示します。このプロンプトに応じて、**ospf\_monitor** のいずれかの対話式コマンドを入力できます。対話式コマンドは、キーボード割り込みでいつでも割り込むことが可能です。

注：コマンド行の長さは、200 文字より短くしてください。

## ローカル・コマンド

|                                   |                                                                                                                                      |
|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| <b>?</b>                          | ローカル・コマンドおよびそれらの関数をすべて表示します。                                                                                                         |
| <b>?R</b>                         | リモート・コマンドおよびそれらの関数をすべて表示します。                                                                                                         |
| <b>d</b>                          | 構成済み宛先をすべて表示します。このコマンドは、 <i>mon_db_file</i> の中に構成されている潜在的なすべての <b>ospf_monitor</b> コマンド宛先の <i>dest_index</i> 、IP アドレス、およびホスト名を表示します。 |
| <b>h</b>                          | コマンド・ヒストリー・バッファを表示します。この場合、最後の 30 個の対話式コマンドが表示されます。                                                                                  |
| <b>x</b>                          | <b>ospf_monitor</b> プログラムを終了します。                                                                                                     |
| <b>@ remote_command</b>           | <i>remote_command</i> を同じ（直前の）宛先に送信します。                                                                                              |
| <b>@dest_index remote_command</b> | <i>remote_command</i> を構成済み宛先 <i>dest_index</i> に送信します。                                                                              |
| <b>F filename</b>                 | すべての <b>ospf_monitor</b> 出力を <i>filename</i> に送信します。                                                                                 |
| <b>S</b>                          | すべての <b>ospf_monitor</b> 出力を stdout に送信します。                                                                                          |

## リモート・コマンド

**a** *area\_id type ls\_id adv\_rtr*

リンク状態通知を表示します。*area\_id* は、照会の送信先である OSPF 領域です。*adv\_rtr* は、このリンク状態通知の発信元であるルーターのルーター ID です。*Type* は、要求する通知のタイプを指定します。これは、次のように指定する必要があります。

- 1 ルーター・リンク通知を要求します。これらの通知は、ルーターのインターフェースの収集済み状態を記述します。このタイプの要求の場合、*ls\_id* フィールドを発信元ルーターのルーター ID に設定する必要があります。
- 2 ネットワーク・リンク通知を要求します。これらの通知は、ネットワークに接続されている一連のルーターを記述します。このタイプの要求の場合、*ls\_id* フィールドをネットワークの指定ルーターの IP インターフェース・アドレスに設定する必要があります。
- 3 ネットワークへの経路を記述している要約リンク通知を要求します。これらの通知は、領域間経路を記述し、領域ボーダーのルーティング情報を圧縮できるようにします。このタイプの要求の場合、*ls\_id* フィールドを宛先ネットワークの IP アドレスに設定する必要があります。
- 4 AS 境界ルーターへの経路を記述している要約リンク通知を要求します。これらの通知は、領域間経路を記述し、領域ボーダーのルーティング情報を圧縮できるようにします。このタイプの要求の場合、*ls\_id* フィールドを記述されている AS 境界ルーターのルーター ID に設定する必要があります。
- 5 AS 外部リンク通知を要求します。これらの通知は、自律システムの外部である宛先への経路を記述します。このタイプの要求の場合、*ls\_id* フィールドを宛先ネットワークの IP アドレスに設定する必要があります。

**c**

累積ログを表示します。このログには、モニター要求、ハロー、データベース記述、リンク状態要求、リンク状態更新、およびリンク状態肯定応答パケットの入出力統計情報が含まれます。経路指定隣接局の合計数およびアクティブな OSPF インターフェースの数を示す領域統計情報が提供されます。経路指定テーブル統計情報は、領域内経路、領域間経路、および AS 外部データベース・エントリーの数として合計されて報告されます。

**e**

累積エラーを表示します。このログは、OSPF 経路指定隣接局相互間に発生する可能性のあるさまざまなエラー状態を報告し、それぞれのエラー状態ごとに出現回数を示します。

**h**

ネクスト・ホップ・リストを表示します。これは、大部分が SPF の計算で出される有効なネクスト・ホップのリストです。

**l** [ *retrans* ]

リンク状態データベースを表示します (ASE の場合は除きます)。このテーブルは、AS を構成しているルーターとネットワークを記述します。*retrans* がゼロ以外であれば、この lsdب 構造体で保持している隣接局の再伝送リストが表示されます。

**A** [ *retrans* ]

AS 外部データベース・エントリーを表示します。このテーブルは、それぞれの AS 外部ルーターごとに、通知ルーター、転送先アドレス、経過時間、長さ、シーケンス番号、タイプ、および経路距離を報告します。*retrans* がゼロ以外であれば、この lsdب 構造体で保持している隣接局の再伝送リストが表示されます。

o [ which ]

OSPF 経路指定テーブルを表示します。このテーブルは、現在 OSPF を介して管理されている AS ボーダー経路、領域ボーダー経路、要約 AS ボーダー経路、ネットワーク、要約ネットワーク、および AS 外部ネットワークを報告します。*which* を省略すると、上記のすべてがリストされます。*which* の値 (1 から 63) を指定すると、ある一定のテーブルだけが表示されます。適切な値は、必要なテーブルの値を合計することで決定されます。値に関しては、次のリストを参照してください。

- 1 この領域内の AS ボーダー・ルーターへの経路。
- 2 この領域に関する領域ボーダー・ルーターへの経路。
- 4 他の領域内の AS ボーダー・ルーターへの要約経路。
- 8 この領域内のネットワークへの経路。
- 16 他の領域内のネットワークへの経路。
- 32 非 OSPF ネットワークへの AS 経路。

I

すべてのインターフェースを表示します。このレポートは、OSPF に対して構成されているすべてのインターフェースを示します。報告される情報には、ネットワークの領域、インターフェース IP アドレス、インターフェース・タイプ、インターフェース状態、コスト、優先順位、および DR と BDR の IP アドレスが含まれます。

N

すべての OSPF 経路指定隣接局を表示します。報告される情報には、領域、ローカル IP アドレス、ルーター ID、隣接局 IP アドレス、状態、およびモードが含まれます。

V

ゲート・バージョン情報を表示します。

## 関連情報

gated デーモン。

---

## pac コマンド

### 目的

プリンターおよびプロッターのアカウント・レコードを作成します。

### 構文

```
/usr/sbin/pac [ -c ] [ -m ] [ -pPrice ] [ -PPrinter ] [ -qFile ] [ -r ] [ -s ] [ Name ... ]
```

### 説明

**pac** コマンドは、選択されたプリンターのユーザーごとに、または、*Name* パラメーターで指定されたユーザーについて、プリンター/プロッターのアカウント・レコードを作成します。プリンターの選択については、**-P** フラグのセクションを参照してください。

計測単位はページ数です。ラスター・デバイスの場合は用紙の長さがフィート単位で計測されます。出力は、使用単位数とドル単位の料金の両方で表されます。1 単位当たりの料金 (価格) については **-p** フラグのセクションを参照してください。

**/etc/qconfig** ファイルに指定したアカウント・ファイルおよび要約情報を入れるために作成したファイルは、root ユーザーまたは **printq** グループに対して読み取りおよび書き込み権限を与える必要があります。

**pac** コマンドは、**/etc/qconfig** ファイルに `acctfile =` 文節で指定したパス名に `_sum` を付加することによって要約ファイル名を生成します。例えば **qconfig** ファイルが以下のものを読むと、  
`acctfile = /var/adm/1p0acct`

**pac** コマンドは要約ファイルに **/var/adm/1p0acct\_sum** という名前を付けます。

## フラグ

- c** ユーザーのアルファベット順の代わりに、価格別に出力をソートします。
- m** ホスト・コンピューターに関係なくユーザーのすべての印刷料金をグループ化します。
- pPrice** 出力の 1 単位当たりで請求される価格をドルで指定します。デフォルトでは、システムは 1 単位当たり \$0.02 を請求します。
- PPrinter** アカウンティング・レコードが作成されるプリンターを指定します。デフォルトでは、システムは **PRINTER** 環境変数またはデフォルト値 **lp0** によって指定されるプリンターを選択します。  
注: **LPDEST** 環境変数を設定すると、同じに機能する **PRINTER** 環境変数より優先されます。コマンド行から出力先オプションを入力すると、**LPDEST** と **PRINTER** の両方の環境変数が上書きされます。
- qFile** キュー構成ファイルを指定します。デフォルト値は **/etc/qconfig** ファイルです。
- r** ソート順序を反転させます。これを指定すると、レコードが z から a までの逆アルファベット順にソートされたり、価格が降順にソートされます。
- s** 要約ファイル内でアカウント情報を要約します。このフラグは処理量の大きいシステムに必要です。

## 例

1. `lp0` プリンターのすべてのユーザーについてプリンター/プロッターのアカウント情報を生成するには、以下のように入力します。

```
/usr/sbin/pac
```

コマンドは印刷されたページ数とユーザーによってソートされた料金を表示します。この例では **PRINTER** 環境変数を指定しないと想定しています。

2. 要約ファイル内のプリンター/プロッターのアカウント・レコードを収集するには、以下のように入力します。

```
/usr/sbin/pac -s
```

3. `lp12` プリンターから `smith`、`jones`、`greene` に関するプリンター/プロッターのアカウント情報を生成するには、以下のように入力します。

```
/usr/sbin/pac -Plp12 smith jones greene
```

注: フラグと変数の間にはスペースを入れないでください。例えば **-pPrice**、**-PPrinter**、**-qFile** のように指定します。

## ファイル

|                      |                         |
|----------------------|-------------------------|
| <b>/usr/sbin/pac</b> | <b>pac</b> コマンドが入っています。 |
| <b>/etc/qconfig</b>  | ファイルへのパスを指定します。         |

## 関連情報

**acctcms** コマンド、**acctcom** コマンド、**acctcon1** または **acctcon2** コマンド、**acctmerg** コマンド、**acctprc1**、**acctprc2**、または **accton** コマンド、**runacct** コマンド。

**qconfig** ファイル。

オペレーティング・システムおよびデバイスのマネージのアカウントिंग・サブシステムの設定のセクション。

オペレーティング・システムおよびデバイスのマネージのシステム・アカウントिंगのセクション。

---

## pack コマンド

### 目的

ファイルを圧縮します。

### 構文

```
pack [ -f ] [ - ] File ...
```

### 説明

**pack** コマンドは、*File* パラメーターで指定したファイルを圧縮したフォーマットで格納します。入力ファイルは、元のファイル名に接尾部 **.z** が付いたパック・ファイルに置き換えられます。呼び出そうとするプロセスに適切な特権がある場合、パック・ファイルのアクセス・モード、アクセス日と修正日、およびオーナーは、元のファイルと同じ値に保たれます。入力ファイル名は付加する接尾部 **.z** のスペースを考慮して 253 バイト以下とします。**pack** コマンドが正常終了すると、元のファイルは除去されます。パックしたファイルは、**unpack** コマンドまたは **pcat** コマンドを使って元のフォーマットに復元することができます。

**pack** コマンドが小型化したファイルを作成できないと、処理を中止して、スペースを節約できないことを報告します (一般に、小さなファイルや均等な文字配置のファイルではスペースを節約できません)。節約されるスペース量は、入力ファイルのサイズと文字の頻度分布によって決まります。デコード・ツリーは各 **.z** ファイルの最初の部分を形成するので、3 ブロックよりも小さいファイルの場合はスペースを節約しません。一般に、テキスト・ファイルは 25 ないし 40% 縮小されます。

**注:** **pack** コマンドのアルゴリズムには、ファイルの圧縮用にハフマン・エンコードが使用されています。このアルゴリズムには根本的な制限があります。したがって、**pack** コマンドでは 8MB 以下のファイルしか一貫して圧縮できません。8MB より大きいファイルを一貫して圧縮するには、**compress** コマンドを使用します。

**pack** コマンドの終了値は、パックできなかったファイルの数です。**pack** コマンドは以下の条件下ではパックを実行しません。

- ファイルが既にパックされている場合。
- 入力ファイル名が 253 バイトを超える場合。
- ファイルがリンクされている場合。
- ファイルがディレクトリーである場合。
- ファイルをオープンできない場合。

- パッキングで節約される記憶ブロックがない場合。
- *File.z* というファイルが既に存在する場合。
- *.z* ファイルを作成できない場合。
- 処理中に入出力エラーが発生した場合。

## フラグ

**-f** *File* パラメーターで指定したファイルのパッキングを強制します。このフラグは、ファイルによって効果がな  
いものもありますが、ディレクトリー全体をバックするときは便利です。

## パラメーター

*File* パックするファイルを指定します。

- *File* パラメーターで指定したファイルについての統計情報を表示します。統計情報は、1 バイトごとに構築さ  
れた Huffman 最小冗長コード・ツリーから計算されます。コマンド行でさらに - (マイナス符号) パラメータ  
ーを指定すると、次に指定されたファイルに対してこの機能が切り替えられます。例 2 を参照してくださ  
い。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0 ファイルが正常にパックされたことを示します。
- >0 エラーが発生したことを示します。

## 例

1. ファイル *chap1* と *chap2* を圧縮して、変更したファイル名を表示するには、以下のように入力しま  
す。

```
pack chap1 chap2
```

圧縮されたバージョンの名前は、*chap1.z* および *chap2.z* に変更されます。**pack** コマンドは、圧縮さ  
れたファイルごとにサイズの縮小率を表示します。

2. 圧縮された量についての統計情報を表示するには、以下のように入力します。

```
pack - chap1 - chap2
```

これにより、ファイル *chap1* および *chap2* が圧縮され、ファイル *chap1* についての統計情報は表示さ  
れますが、*chap2* の統計情報は表示されません。最初の - (マイナス符号) パラメーターは統計情報の表  
示をオンにし、2 番目の - パラメーターはオフにします。

## ファイル

**/usr/bin/pack** **pack** コマンドが入っています。

## 関連情報

**cat** コマンド、**pcat** コマンド、**unpack** コマンド。

オペレーティング・システムおよびデバイスのマネージのファイルのセクション。

オペレーティング・システムおよびデバイスのマネージの入出力ダイレクトのセクション。

---

## packf コマンド

### 目的

フォルダーの内容を圧縮して 1 つのファイルに入れます。

### 構文

**packf** [ *+Folder* ] [ *Messages* ] [ **-file** *File* ]

### 説明

**packf** コマンドは、フォルダー内のメッセージを指定するファイルに圧縮します。デフォルトでは、**packf** コマンドは現行フォルダーのメッセージを圧縮して **msgbox** ファイルに入れます。このファイルが存在しない場合、システムはファイルを作成するかどうか確認するプロンプトを表示します。ファイルの各メッセージは、4 つの Ctrl-A 文字と改行文字 1 文字で区切ります。

**注:** **inc** コマンドを使用して、圧縮したメッセージをアンパックすることができます。

### フラグ

**-file** *File*

圧縮したメッセージを入れるファイルを指定します。デフォルトは **./msgbox** ファイルです。このファイルが存在する場合は、**packf** コマンドはメッセージをファイルの最後に追加します。このファイルが存在しない場合、システムはファイルを作成するかどうか確認するプロンプトを表示します。

**+Folder**

圧縮したいメッセージが入っているフォルダーを示します。

**-help**

コマンド構文、使用可能なスイッチ (トグル)、およびバージョン情報をリストします。

**注:** メッセージ・ハンドラー (MH) の場合は、このフラグの名前は完全につづらなければなりません。



## Messages

圧縮するメッセージを指定します。 *Messages* パラメーターには、複数のメッセージ、メッセージの範囲、または単一のメッセージを指定できます。複数のメッセージを指定すると、最初に圧縮されたメッセージが現行メッセージになります。以下のリファレンスを使用してメッセージを指定します。

*Number* メッセージの数。複数のメッセージを指定するときは、それぞれの番号をスペースで区切ります。範囲を指定するときは、その範囲の最初と最後の値をハイフンで区切ります。

### Sequence

ユーザーが指定するメッセージのグループ。指定できる値は、以下のとおりです。

**all** フォルダー内のすべてのメッセージ。これはデフォルトです。

**cur** または **.** (ピリオド)  
現行メッセージ。

**first** フォルダー内の最初のメッセージ。

**last** フォルダー内の最後のメッセージ。

**next** 現行メッセージの次のメッセージ。

**prev** 現行メッセージの前のメッセージ。

## プロファイル・エントリー

下記のエントリーは、*UserMhDirectory.mh\_profile* ファイルに入力します。

Current-Folder: デフォルトの現行フォルダーを設定します。  
Msg-Protect: 新しいメッセージ・ファイルの保護レベルを設定します。  
Path: ユーザーの MH ディレクトリーを指定します。

## 例

1. 現行フォルダーのすべてのメッセージを圧縮し、その結果のテキストを **schedule** ファイルに入れるには、以下のように入力します。

```
packf -file schedule
```

システムは次のようなメッセージで応答します。

```
Create file "/home/mary/schedule"?
```

y を入力するとファイルが作成されます。

2. **test** フォルダーの 3 から 7 の範囲のメッセージを既存の **msgbox** ファイルに圧縮するには、以下のように入力します。

```
packf +test 3-7
```

システムは、コマンドが完了すると shell プロンプトを表示します。

3. **inbox** フォルダーの現行メッセージ、最初のメッセージ、最後のメッセージを既存の **msgbox** ファイルに圧縮するには、以下のように入力します。

```
packf cur first last
```

## ファイル

`$HOME/mh_profile`  
`/usr/bin/packf`

MH ユーザー・プロファイルを指定します。  
**packf** コマンドが入っています。

## 関連情報

**inc** コマンド。

**.mh\_alias** ファイル・フォーマット、**.mh\_profile** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ のメール・アプリケーションのセクション。

---

## pagdel コマンド

### 目的

現行プロセスのクレデンシャルに存在するすべての PAG 関連付けを除去します。

### 構文

**paginit** [ **-R** *module\_name* ] [ *username* ]

### 説明

**pagdel** コマンドは、現行プロセスのクレデンシャル構造から PAG ID を除去します。**-R** オプションを省略すると、レジストリー属性が **module\_name** として使用されます。

### フラグ

**-R** *module\_name* **/usr/lib/security/modules.cfg** にあるロード・モジュールを指定します。**load\_module** は、現在プロセスに関連付けられているすべての PAG を削除することを求められます。

## セキュリティ

アクセス制御: このコマンドは、実行 (x) アクセス権を root ユーザーとセキュリティ・グループのメンバーだけに与えます。このコマンドは、トラステッド・コンピューティング・ベース (TCB) 内のプログラムとしてインストールします。このコマンドは、**setuid** (SUID) ビットが設定されている root ユーザーが所有しなければなりません。

### 監査

USER\_PagDelete

### 例

現行プロセスに関連付けられた PKI 認証証明を除去するには、次のように入力します。

```
pagdel -R FPKI
```

## 関連情報

**paginit** および **paglist** コマンド。

---

## pagesize コマンド

### 目的

システムのページ・サイズを表示します。

### 構文

```
pagesize [ -a ] [ -f ]
```

### 説明

**pagesize** コマンドは、**getpagesize** サブルーチンによって戻されるメモリーのページ・サイズをバイト単位で出力します。このコマンドはポータブル・シェル・スクリプトを作成する場合にシステムの互換性を実現するのに役立ちます。

**-a** フラグを指定すると、**pagesize** コマンドはシステム上でサポートされるすべてのページ・サイズ値 (単位：バイト) を印刷します。

### フラグ

- a** システム上でサポートされるすべてのページ・サイズ値 (単位：バイト) を印刷します。
- f** バイト表記のページ・サイズではなく、英字のサフィックスが付いたフォーマット済みページ・サイズ (例えば、4K) を印刷します。

### 例

1. システム・ページ・サイズを調べるには、以下のように入力します。

```
pagesize
```

システムは、例えば 4096 などバイト単位の数値を戻します。

2. フォーマット済みのページ・サイズを印刷するには、以下のように入力します。

```
pagesize -f
```

システムはフォーマット済みのページ・サイズ (例えば、4K) を戻します。

3. サポートされるページ・サイズすべてを英字サフィックス付きで印刷するには、以下のように入力します。

```
pagesize -af
```

システムはサポートされるすべてのページ・サイズを戻します。以下に例を示します。

```
4K
64K
16M
```

### ファイル

`/usr/bin/pagesize` **pagesize** コマンドが入っています。

### 関連情報

**getpagesize** サブルーチン。

---

## paginit コマンド

### 目的

ユーザーを認証し、PAG 関連付けを作成します。

### 構文

```
paginit [ -R module_name ] [ username ]
```

### 説明

**paginit** コマンドは、*username* (デフォルトでは、コマンドを実行するユーザー) を認証し、*username* とカーネル・トークン (プロセス認証グループ・エントリ (PAG) と呼ばれる) の関連付けを作成します。このコマンドにより、新しいログイン・シェルが作成されます。

**-R** フラグを指定しないと、**paglist** はユーザーのレジストリー属性を照会し、その値を *module\_name* に使用します。

*username* を代替の Identification and Authentication (I&A) メカニズムと関連付けるために、**-R** フラグを使用して、ユーザーの作成に使用する I&A ロード・モジュールを指定することができます。ロード・モジュールは、**/usr/lib/security/methods.cfg** ファイルに定義されます。

### フラグ

**-R** *module\_name*           ユーザーの認証に使用するロード可能な I&A モジュールを指定します。

### パラメーター

*username*                   ユーザーを指定します。このパラメーターのデフォルトは、コマンドを実行するユーザーです。**root** ユーザーのみがデフォルトを変更できます。

### セキュリティ

アクセス制御: このコマンドは誰もが実行できるようにすべきです。これは **root** により所有され、**setuid** されるべきです。

### 監査

USER\_Paginit

### 例

```
paginit -R FPKI
```

ユーザーはレジストリー **FPKI** を使用して認証され、これは **/usr/lib/security/methods.cfg** ファイルに定義されています。PAG は現行のプロセス・クレデンシャルと関連付けられます。

### 関連情報

**pagdel** および **paglist** コマンド。

---

## paglist コマンド

### 目的

現行プロセスに関連付けられた認証情報をリストします。

### 構文

```
paglist [ -R module_name ]
```

### 説明

**paglist** コマンドは現行プロセスのクレデンシャルを照会し、その認証証明書を表示します。

**-R** オプションを指定しないと、**paglist** はユーザーのレジストリー属性を照会し、その値を **module\_name** に使用します。

### フラグ

**-R *module\_name***          ロード・モジュール *module\_name* が現行プロセスに関連付けられた認証証明をリストすることを指定します。

## セキュリティ

アクセス制御: このコマンドは、このコマンドを呼び出したユーザーの ID を使用して実行され、それ以上の特権は必要ありません。これは root が所有すべきですが、誰もが実行できるべきものです。

### 例

```
paglist -R FPKI
```

この例は、FPKI レジストリー内の現行プロセスに関連付けられた PAG をリストします。

### 関連情報

**pagdel** および **paginit** コマンド。

---

## panel20 コマンド

### 目的

HIA と 5080 制御装置の間のアクティビティを診断します。

### 構文

```
panel20 [ HIA0 | HIA1 | HIA2 ]
```

### 説明

**panel20** コマンドを診断ツールとして使用して、ホスト・インターフェース・アダプター (HIA) が正しくインストールされ、5088 Graphics Channel Control Unit (GCCU) と通信しているかどうかを調べます。

**panel20** コマンドは、以下のカラムのある診断画面を表示します。 Device Name、Channel Address、Link Address、Link Status、Poll Counter、SNRM Counter。

HIA が正しくインストールされ、ホスト・オペレーティング・システムが 5088 上で 3270 デバイスをサポートするように正しく構成されている場合は、Set Normal Response Mode (SNRM Counter) カラムのエントリーは増えます。SNRM Counter のエントリーが増えない場合は、HIA の問題判別手順を参照し、ホスト・オペレーティング・システムが正しく構成されていることを確認します。

## 例

**panel20** コマンドを開始するには、以下のように入力します。

```
panel20
```

デフォルトでは、**panel20** コマンドは HIA0 をモニターします。HIA1 または HIA2 をモニターするには、以下のように入力します。

```
panel20 HIA1
```

または

```
panel20 HIA2
```

---

## passwd コマンド

### 目的

ユーザーのパスワードを変更します。

### 構文

```
passwd [ -R load_module ] [ -f | -s ] [ User ]
```

### 説明

**passwd** コマンドは、ユーザーのパスワードの設定と変更を行います。このコマンドを使用して、自分のパスワードまたはほかのユーザーのパスワードを変更します。**passwd** コマンドを使用して、ログイン名と関連した完全名 (gecos) と、オペレーティング・システムへのインターフェースとして使用するシェルを変更することもできます。

ユーザーの定義方法によって、そのユーザーのパスワードをローカルまたはリモートにすることができません。ローカル・パスワードは **/etc/security/passwd** データベースにあります。リモート・パスワードは、リモート・ドメインによって提供されるデータベースに保管されます。

自分のパスワードを変更するには、**passwd** コマンドを入力します。**passwd** コマンドは、root ユーザーではないユーザーに旧パスワード (ある場合) の入力を求めるプロンプトを表示し、次に新規パスワードの入力を求めるプロンプトを 2 回表示します。(パスワードは画面には表示されません。) 新規パスワードの 2 つの入力が一致しないと、**passwd** コマンドは新規パスワードの入力を求めるプロンプトを再度表示します。

**注:** **passwd** コマンドは、ローカル・パスワードおよび NIS パスワードの場合、パスワードの最初の 8 文字だけを使用します。7 ビット文字だけがパスワードでサポートされます。このため、各国語サポート (NLS) コード・ポイントはパスワードでは許可されません。

ほかのユーザーのパスワードを変更するには、**passwd** コマンドとそのユーザーのログイン名 (*User* パラメーター) を入力します。ほかのユーザーのパスワードの変更を許可されているのは、**root** ユーザーか、セキュリティ・グループのメンバーだけです。**passwd** コマンドは、新規パスワードのほかにそのユーザーの旧パスワードの入力を求めるプロンプトを表示します。ローカル・パスワードの場合、**passwd** コマンドは、**root** ユーザーに旧ユーザー・パスワードまたは **root** パスワードの入力を求めません。リモート・パスワードの場合、デフォルトによって **root** ユーザーには旧パスワードの入力が求められ、それにより、リモート・ドメインがそのパスワードを使用するか無視するかを決定できます。この動作を変更するには、**/usr/lib/security/methods.cfg** ファイルの **rootrequiresopw** オプションを参照してください。**passwd** コマンドは、**root** ユーザーにはパスワード制限を強制しません。

**/etc/passwd** ファイルは、完全名と、使用するシェルのパス名を記録します。記録された名前を変更するには、**passwd -f** コマンドを入力します。ログイン・シェルを変更するには、**passwd -s** コマンドを入力します。

**/etc/security/user** 構成ファイルのパスワード制限に従って、ローカルに定義されたパスワードを作成します。このファイルには、以下の制限が入っています。

|                    |                                                                                                                      |
|--------------------|----------------------------------------------------------------------------------------------------------------------|
| <b>minalpha</b>    | 英字の最小数を指定します。                                                                                                        |
| <b>minother</b>    | その他の文字の最小数を指定します。                                                                                                    |
| <b>minlen</b>      | 文字の最小数を指定します。<br>注: この値は、 <b>minalpha</b> に <b>minother</b> の値を足したものが、または <b>minlen</b> の値のいずれか大きい方によって決まります。        |
| <b>mindiff</b>     | 旧パスワードになく、新規パスワードにある文字の最小数を指定します。<br>注: この制限では位置は関係ありません。新規パスワードが <b>abcd</b> で旧パスワードが <b>edcb</b> の場合、異なる文字数は 1 です。 |
| <b>maxrepeats</b>  | ある 1 つの文字をパスワード内で使用できる最大回数を指定します。                                                                                    |
| <b>minage</b>      | パスワードを変更できる最低経過日数を指定します。パスワードは、最低期間が経過するまで変更できません。この値は週単位で計測します。                                                     |
| <b>maxage</b>      | パスワードの最長経過日数を指定します。パスワードは、週単位で計測する指定された期間経過後に変更しなくてはなりません。                                                           |
| <b>maxexpired</b>  | ユーザーがパスワードを変更できる期間で、 <b>maxage</b> を超える期間の最大値を週単位で指定します。                                                             |
| <b>histexpire</b>  | ユーザーがパスワードを再利用できない期間の値を週単位で指定します。                                                                                    |
| <b>histsize</b>    | ユーザーが再利用できない以前のパスワードの数を指定します。                                                                                        |
| <b>dictionlist</b> | パスワードが変更されたときに検査されるディクショナリー・ファイルのリストを指定します。                                                                          |
| <b>pwdchecks</b>   | パスワードが変更されると呼び出される外部パスワード制限メソッドのリストを指定します。                                                                           |

**root** ユーザーが **NOCHECK** 属性を **/etc/security/passwd** ファイルのフラグ・エントリーに追加した場合は、パスワードがこれらの制限を満たしていなくてもかまいません。また、**root** ユーザーは、以下のパスワード制限に従わずにほかのユーザーに新規パスワードを割り当てることができます。

**root** ユーザーが **ADMIN** 属性をフラグ・エントリーに追加するか、または **/etc/passwd** ファイルの **password** フィールドに \* (アスタリスク) が含まれている場合は、**root** ユーザーだけがパスワードを変更できます。また、**/etc/passwd** の **password** フィールドに ! (感嘆符) が含まれ、かつ **/etc/security/passwd** ファイルの **password** フィールドに \* が含まれる場合、**root** ユーザーだけがユーザーのパスワードを変更する排他的特権を持ちます。

**root** ユーザーがパスワードを変更すると、**ADMCHG** 属性が自動的に **/etc/security/passwd** ファイルのフラグ・エントリーに追加されます。この場合は、次にログインするときにパスワードを変更しなくてはなりません。



**/etc/security/user** ファイルのユーザーの **registry** の値が DCE または NIS のいずれかである場合、パスワード変更は指定されたデータベースでだけ行うことができます。

## フラグ

|                              |                                                                                             |
|------------------------------|---------------------------------------------------------------------------------------------|
| <b>-f</b>                    | <b>finger</b> コマンドがアクセスするユーザー情報を変更します。このフラグを使用して、 <b>/etc/passwd</b> ファイルにフルネームを入れることができます。 |
| <b>-s</b>                    | ログイン・シェルを変更します。                                                                             |
| <b>-R</b> <i>load_module</i> | ユーザーのパスワードを変更するために使用されるロード可能 I&A モジュールを指定します。                                               |

## セキュリティ

**passwd** コマンドは、**passwd** というサービス名を持つ、PAM 使用可能アプリケーションです。認証に PAM を使用するシステム規模の構成は、**/etc/security/login.cfg** の **usw** スタンザにある **auth\_type** 属性の値を、**root** ユーザーと同じ **PAM\_AUTH** に変更することにより、設定されます。

PAM が使用可能であるときに使用される認証メカニズムは、**/etc/pam.conf** 内の **passwd** サービスの構成によって決まります。**passwd** コマンドには、パスワード・モジュール・タイプとして **/etc/pam.conf** エントリが必要です。以下にリストされるのは、**passwd** サービスにおける **/etc/pam.conf** 内の推奨される構成です。

```
#  
# AIX passwd configuration  
#  
passwd password required /usr/lib/security/pam_aix
```

## 例

1. パスワードを変更するには、以下のように入力します。

```
passwd
```

**passwd** コマンドは、旧パスワードがあり、ユーザーが **root** 以外の場合は、旧パスワードの入力を求めるプロンプトを表示します。旧パスワードを入力すると、新規パスワードの入力を求めるプロンプトが 2 回表示されます。

2. **/etc/passwd** ファイルのフルネームを変更するには、以下のように入力します。

```
passwd -f
```

**passwd** コマンドは、そのユーザー ID に格納されている名前を表示します。例えばログイン名 **sam** の場合、**passwd** コマンドは次のようなメッセージを表示します。

```
sam's current gecoc:  
    "Sam Smith"  
Change (yes) or no)? >
```

はい (yes) の場合 **Y** を入力すると、**passwd** コマンドは新しい名前を入力を求めるプロンプトを表示します。**passwd** コマンドは、**/etc/passwd** ファイルに入力した名前を記録します。

3. 次にログインするとき異なるシェルを使用するには、以下のように入力します。

```
passwd -s
```

**passwd** コマンドは、使用可能なシェルのパス名と、現在使用中のシェルをリストします。次のようなプロンプトも表示します。

```
Change (yes) or (no)? >
```

はい (yes) の場合 Y を入力すると、**passwd** コマンドは使用するシェルの入力を求めるプロンプトを表示します。次にログインするとき、システムはここで指定したシェルを提供します。

## ファイル

**/usr/bin/passwd**

**/etc/passwd**

**/etc/security/passwd**

**passwd** コマンドが入っています。

ユーザー ID、ユーザー名、ホーム・ディレクトリー、ログイン・シェル、およびフィンガー情報が入っています。

暗号化されたパスワードとセキュリティー情報が入っています。

## 関連情報

**chfn** コマンド、**chsh** コマンド、**login** コマンド、**pwdadm** コマンド、**pwdck** コマンド。

**getpass** サブルーチン、**newpass** サブルーチン。

セキュリティーのネットワークの保護のセクション。

セキュリティーのトラステッド・コンピューティング・ベースのセクション。

使用可能なシェルの詳細は、オペレーティング・システムおよびデバイスのマネージのシェルのセクション。

---

## paste コマンド

### 目的

異なるファイルの行を結合します。

### 構文

```
paste [ -s ] [ -d List] File1...
```

### 説明

**paste** コマンドは、コマンド行に指定したファイルから入力を読みます。 - (マイナス符号) がファイル名として表示されると、標準入力から読みます。このコマンドは指定した入力ファイルの対応する行を連結し、その結果の行を標準出力に書き出します。

デフォルトでは、**paste** コマンドは各ファイルを 1 列として処理し、これらのファイルをタブ文字を使って横に結合します (パラレル・マージ)。**paste** コマンドは、**cat** コマンドに対応するものと考えられます (**cat** コマンドはファイルを縦に、つまりファイルを 1 つずつ結合します)。

**-s** フラグを指定すると、**paste** コマンドは同じ入力ファイルの後続の行を結合します (シリアル・マージ)。デフォルトでは、タブ文字で行を結合します。

注:

1. **paste** コマンドは、32767 個までの入力ファイルをサポートします (**OPEN\_MAX** 定数)。

2. **pr -t -m** コマンドのアクションは **paste** コマンドと同様ですが、ページ・レイアウトを整えるために、スペース、タブ、行を追加作成します。
3. 入力ファイルはテキスト・ファイルでなくてはなりません、行の長さは無制限です。

## フラグ

**-d List** 出力内で対応する行を区切る区切り文字を、*List* パラメーター内で指定した 1 つまたは複数の文字で変更します (デフォルトはタブです)。 *List* パラメーター内に複数の文字があると、それらの文字がその順序で出力の終わりまで反復されます。パラレル・マージの場合、最後のファイルの行は、*List* パラメーターの文字でなく常に改行文字で終了します。

*List* パラメーター内では以下の特殊文字も使えます。

|           |                    |
|-----------|--------------------|
| <b>\n</b> | 改行文字               |
| <b>\t</b> | タブ                 |
| <b>\\</b> | バックスラッシュ           |
| <b>\0</b> | 空文字列 (null 文字ではない) |
| <b>c</b>  | 拡張文字               |

シェルに対して特別な意味を持つ文字は引用符で囲まなくてはなりません。

**-s** 最初のファイルの後続の行を横にマージします。このフラグを指定すると、**paste** コマンドは次のファイルに開始する前に 1 つのファイル全体に処理します。1 つのファイルで行のマージが終了すると、改行を強制し、それから次の入力ファイルの行をマージするといったやり方で、残りの入力ファイルを一度に 1 つずつ処理します。**-d** フラグを指定する場合以外は、タブで行を区切ります。*List* パラメーターにかかわらず、ファイルの最後の文字は必ず改行文字になります。

## 終了状況

このコマンドは、以下の終了値を戻します。

**0** 正常終了。  
**>0** エラーが発生しました。

## 例

1. データの複数の列を一度にペーストするには、以下のように入力します。

```
paste names places dates > npd
```

`npd` というファイルが作成されます。このファイルでは、ある列に `names` ファイル、別の列に `places` ファイル、さらに別の列に `dates` ファイルのデータが入っています。`names`、`places`、`dates` ファイルが以下のような場合、

| names  | places     | dates      |
|--------|------------|------------|
| rachel | New York   | February 5 |
| jerry  | Austin     | March 13   |
| mark   | Chicago    | June 21    |
| marsha | Boca Raton | July 16    |
| scott  | Seattle    | November 4 |

`npd` ファイルの内容は以下のようになります。

|        |            |            |
|--------|------------|------------|
| rachel | New York   | February 5 |
| jerry  | Austin     | March 13   |
| mark   | Chicago    | June 21    |
| marsha | Boca Raton | July 16    |
| scott  | Seattle    | November 4 |

タブ文字が各行の名前、ロケーション、および日付を区切ります。タブ・ストップは 8 桁目ごとに設定されているので、これらの列が必ずしも位置調整されるとは限りません。

2. 列をタブ以外の文字で区切るには、以下のように入力します。

```
paste -d"!@" names places dates > npd
```

上記の例では、! と @ を列セパレーターとして交互に使用します。names、places、dates ファイルが例 1 と同じであれば、npd ファイルの内容は以下のようになります。

```
rachel!New York@February 5
jerry!Austin@March 13
mark!Chicago@June 21
marsha!Boca Raton@July 16
scott!Seattle@November 4
```

3. 標準入力を複数の列に表示するには、以下のように入力します。

```
ls | paste - - - -
```

ここでは、現行ディレクトリーが 4 つの列にリストされます。- (マイナス) はそれぞれ **paste** コマンドに、標準入力から読んだデータを含む列を作成するように指示します。1 行目が 1 列目に、2 行目が 2 列目にとように入れます。

これは次のコマンドと等価です。

```
ls | paste -d"\t\t\t\n" -s -
```

この例では、標準入力から得られた後続行によって、このページにある列が埋められます。

-d"\t\t\t\n" は各列の後に挿入される文字を定義します。タブ文字 (\t) は最初の 3 列の後に、改行文字 (\n) は 4 列目の後に挿入されます。-d フラグを指定しないと、**paste -s -** コマンドはすべての入力を 1 行として表示し、各列の間をタブ文字で区切ります。

## ファイル

/usr/bin/paste            **paste** コマンドが入っています。

## 関連情報

**cat** コマンド、**cut** コマンド、**grep** コマンド、**pr** コマンド。

*AIX 5L* バージョン 5.3 ナショナル・ランゲージ・サポート ガイドおよびリファレンスのナショナル・ランゲージ・サポートの概要のセクション。

オペレーティング・システムおよびデバイスのマネージのファイルのセクション。

オペレーティング・システムおよびデバイスのマネージの入出力リダイレクトのセクション。

---

## patch コマンド

### 目的

変更内容をファイルに適用します。

### 構文

```
patch [ -b [ -B Prefix ] ] [ -f ] [ -l ] [ -N ] [ -R ] [ -s ] [ -v ] [ -c | -e | -n ] [ -d  
Directory ] [ -D Define ] [ -F Number ] [ -i PatchFile ] [ -o OutFile ] [ -p Number ] [ -r  
RejectFile ] [ -x Number ] [ File ]
```

### 説明

**patch** コマンドは、ファイルの変更方法についてのソース・ファイルの命令を読み、その変更を適用します。このソース・ファイルには、**diff** コマンドで作成される差分に関するリスト (*diff* リスト) が入っています。*diff* リストは、2 つのファイルを比較し、差分の調整方法に関する命令を構築した結果です。デフォルトでは、**patch** コマンドは標準入力から読んだソース・ファイルを使用しますが、この動作は **-i** フラグと *PatchFile* 変数を指定すると上書きできます。

*diff* リストは、通常フォーマット、コンテキスト・フォーマット、または **ed** エディターのスタイルという 3 つのフォーマットで表示されます。**-c**、**-e**、または **-n** フラグで抑制する場合以外、*diff* リストのフォーマットは **patch** コマンドによって決定されます。

デフォルトでは、ファイルのパッチ・バージョンによって元のバージョンが置き換えられます。**-b** フラグを指定すると、各パッチ・ファイルの元のファイルは、同じファイル名に接尾部 **.orig** が付加されたファイルに保存されます。また、**-o** フラグを指定すると、出力先を指定できます。

**-p** フラグを指定すると、パッチ・ファイルを手作業で編集せずに、ローカル・ユーザーのディレクトリー構造に合わせてパッチ・ファイルをカスタマイズすることができます。そのためには、絶対パス名から除去するコンポーネント数を指定します。例えば、パッチ・ファイルにパス名 */curds/whey/src/blurfl/blurfl.c* が入っているとすると、

- **-p 0** を指定すると、パス名全体が使用されます。
- **-p 1** を指定すると、先行スラッシュが除去され、*curds/whey/src/blurfl/blurfl.c* となります。
- **-p 4** を指定すると、先行スラッシュと最初の 3 つのディレクトリーが除去され、*blurfl/blurfl.c* となります。

**-p** フラグを指定しないと、**patch** コマンドはベース名を使用します。上の例では *blurfl.c* となります。

### パッチ・ファイルのフォーマット

パッチ・ファイルには、1 つまたは複数のパッチが後に続く 1 行または複数行のヘッダー情報が入っていません。各パッチには、1 行または複数行のファイル名 ID が **diff -c** コマンドによって作成されたフォーマットで入っていません。1 つまたは複数の **diff** コマンド出力のセットを、通常 *hunk* (ハンク) といいます。

**patch** コマンドは、パッチ・ファイル内の先行テキストをスキップし、実際の *diff* リストを適用し、後続のテキストをスキップします。したがって、*diff* リストを含むエントリーまたはメッセージをパッチ・ファイルとして使用しても、**patch** コマンドは引き続き機能します。その場合、*diff* リスト全体が一定量だけ字下げされていると、**patch** コマンドもそのスペーシングに合わせて調整します。

元のファイル内の行範囲を変更するには、パッチ内の各ハンクが別個の diff リストでなくてはなりません。パッチ内の連続するハンクの行番号は、昇順で指定しなくてはなりません。

## ファイル名の判別

*File* パラメーターを指定しないと、**patch** コマンドは以下の手順を行って、編集するファイルの名前を取得します。

1. コンテキスト diff リストのヘッダーでは、ファイル名は \*\*\* (3 個のアスタリスク) または -- (3 個のダッシュ) で始まる行から判別されます。\*\*\* で始まる行はパッチが取り出されたファイルの名前を示し、-- で始まる行はパッチを適用するファイルの名前を示します。既存のファイルの最も短い名前が選択されます。
2. 先行テキスト内に **Index:** 行があると、**patch** コマンドはその行のファイル名を使おうとします。
3. コンテキスト diff ヘッダーは **Index:** 行よりも優先されます。
4. 先行テキストからファイル名を判別できない場合、**patch** コマンドは、パッチするファイルの名前の入力を求めるプロンプトを表示します。
5. 元のファイルが見つからなくても、適切な SCCS ファイルまたは RCS ファイルが使用可能であれば、**patch** コマンドはそのファイルを取得またはチェックアウトしようとしています。
6. 先行テキストに **Prereq:** 行があると、**patch** コマンドは前提条件の行から最初のワード (通常はバージョン番号) を取り出し、入力ファイルにそのワードがあるかどうかを調べます。見つからない場合は、**patch** コマンドは処理を続ける前に確認を求めるプロンプトを表示します。

## パッチの適用

パッチ・ファイルに複数のパッチが入っていると、**patch** コマンドは各 diff リストを別のパッチ・ファイルから取り出したものとして適用しようとしています。この場合、パッチされるファイルの名前は diff リストごとに判別され、各 diff リストの前に表示されるヘッダー・テキスト内でファイル名や改訂レベルなどの情報が検査されます。

**-c**、**-e**、または **-n** フラグを指定すると、**patch** コマンドは各ハンク内の情報を、それぞれコンテキストの差分、ed エディターの差分、通常の差分として解釈します。これらのフラグを指定しないと、**patch** コマンドはハンク内の情報のフォーマットに基づいて差分のタイプを判別します。

**patch** コマンドは、ハンクの最初の行の番号を取り出して、直前のハンクを適用することによって生じる行オフセットを加算または減算することにより、各ハンクの適用個所を検索します。この行位置で正確に一致させることができない場合は、**patch** コマンドはハンクの内容と正確に一致する行の集合を、順方向と逆方向の両方にスキャンします。

この適用個所が見つからず、**patch** コマンドがコンテキスト diff リストを適用している場合は、**patch** コマンドは、完全ではないが部分的に一致するものを検索することができます。*fuzz* ファクターは、部分的に一致する行がどのくらいあるかを指定します。*fuzz* ファクターを 1 以上の値に設定すると、**patch** コマンドは 2 回目のスキャンを実行します。このときはコンテキストの先頭行と最終行を無視します。一致するものがない場合、最大 *fuzz* ファクターが 2 以上に設定されていると、**patch** コマンドは 3 回目のスキャンを実行します。このときはコンテキストの最初の 2 行と最後の 2 行を無視します (デフォルトの最大 *fuzz* ファクターは 2 です)。これで一致するものがない場合、**patch** コマンドはそのハンクをリジェクト・ファイルに入れます。リジェクト・ファイルは、出力ファイルと同じ名前と接尾部 **.rej** で作成されます。この命名規則は、**-r** フラグを指定すると上書きできます。

リジェクトされたハンクは、パッチ・ファイルのフォーマットに関係なくコンテキスト diff リスト・フォーマットで書かれます。入力が通常または ed エディター・スタイルの差分であれば、リジェクト・ファイルにはコンテキスト・フォーマットの行がない差分が含まれることがあります。リジェクト・ファイル内の



ハンクに関する行番号は、パッチ・ファイル内の行番号とは異なることがあります。これは、リジェクト・ファイルの行番号には、旧ファイルではなく新規ファイル内で失敗したハンクの位置の近似値が反映されるからです。

各ハンクが完了するたびに、**patch** コマンドはハンクが正常に終了したか失敗したかを示します。また、各ハンクに想定された新しい行番号も表示されます。この行番号が **diff** リスト内で指定した行番号と異なる場合は、オフセットが通知されます。また、**patch** コマンドは、**fuzz** ファクターを使用して一致を探したかどうか也表示します。

**注:** 1 つの広範なオフセットが通知されると、ハンクが正しくないロケーションにインストールされた可能性があります。**fuzz** ファクターを使用した場合も、間違った配置を示すことがあります。

## ほかのユーザーのためのパッチの準備

ほかのユーザーに配布するパッチを準備するプログラマーは、ほかにも以下のガイドラインに注意しなくてはなりません。

- 同じパッチを 2 回適用しようとするすると、**patch** コマンドは 2 回目の適用が逆パッチであると想定し、逆パッチの確認を求めるプロンプトを表示します。したがって、逆パッチを適用するとユーザーは既にパッチを適用したかがわかりにくくなるので、逆パッチを送信しないようにしてください。
- 最新のパッチ・レベルで更新される **patchlevel.h** ファイルを保管することをお勧めします。これにより、送信するパッチ・ファイル内の最初の **diff** リストとして、パッチ・レベルを使用できます。パッチに **Prereq:** 行が含まれていると、秩序の正しくないパッチに対しては必ず警告されます。
- コンテキスト **diff** リスト・ヘッダーで、または **Index:** 行で、ファイル名を正しく指定しなくてはなりません。サブディレクトリー内のものをパッチする場合は、必要に応じて **-p** フラグを指定するようにパッチ・ユーザーに必ず指示してください。
- **null** ファイルを作成したいファイルと比較する **diff** リストを送信することによって、ファイルを作成することができます。ただしこの方法は、作成したいファイルがターゲット・ディレクトリー内に存在していない場合にしか使用できません。
- 多数の **diff** リストを 1 つのファイルに入れることは可能ですが、関連するパッチを別々のファイルにグループ化しておくことをお勧めします。
- **patch** コマンドは、**ed** スクリプト内の行番号が正しくないかどうかを表示できません。変更または削除コマンドを見つけたときに、通常 **diff** リスト内の正しくない行番号を示すことだけができます。**fuzz** ファクター 3 を使用するコンテキスト **diff** リストには、同じ行番号の問題が発生することがあります。このような場合には、適切な対話型インターフェースを追加するまでは、コンテキスト **diff** リストを使用して変更が正確かどうかを検査してください。エラーが発生せずにコンパイルされた場合は、一般にパッチが機能したことを意味しますが、パッチ機能を絶対保証できるわけではありません。
- **patch** コマンドの結果は、パッチが、その生成元ファイルと正確に同じバージョンに適用されるときだけ保証されます。
- 以下のようにコードが重複する場合は仮定します。

```
#ifdef
... NEWCODE
#else
... OLDPCODE
# endif
```

**patch** コマンドは、両方のバージョンともパッチできません。**patch** コマンドが正常に終了した場合は、正しくないバージョンをパッチして、正常な終了状況に戻した可能性があります。



## フラグ

- b** 差分が適用される前に、修正される各ファイルのコピーを保存します。コピーされた元のファイルは、同じ名前に接尾部 **.orig** が付いたファイルになります。その名前のファイルが既に存在する場合は上書きされます。複数のパッチを同じファイルに適用すると、最初のパッチの時点で元のファイルのコピーが 1 つだけ作成されます。 **-o OutFile** フラグも指定すると、**.orig** ファイルは作成されません。しかし、指定された out ファイルが既に存在する場合は、**OutFile.orig** が作成されます。
- B Prefix** バックアップ・ファイル名の接頭部を指定します。このフラグは、**-b** フラグと一緒に指定する場合だけ機能します。
- c** パッチ・ファイルをコンテキスト diff リスト (**diff -c** または **diff -C** コマンドの出力) として解釈します。このフラグは **-e** または **-n** フラグと一緒に指定できません。
- d Directory** 処理の前に、現行ディレクトリーを指定したディレクトリーに変更します。
- D Define** 以下の C プリプロセッサ命令で変更箇所をマークします。
- ```
#ifdef Define
... (NEWCODE)
#else
... (OLDCODE)
#endif /* Define */
```
- Define** 変数は差分指定記号として使用されます。このフラグは、通常フォーマットまたはコンテキスト・フォーマットの diff リストがパッチ・ファイルとして使用される場合のみ機能します。
- e** パッチ・ファイルを ed エディター・スクリプトとして解釈します。このフラグは **-c** または **-n** フラグと一緒に指定できません。
- f** ユーザーへの照会を抑制します。コメント表示を抑制するには、**-s** フラグを指定します。
- F Number** 最大 fuzz ファクターを設定します。このフラグはコンテキスト diff リストにのみ適用され、**patch** コマンドは、ハンクのインストール先を判別するとき、指定された行数を無視します。 **-F** フラグを指定しない場合、デフォルトの fuzz ファクターは 2 です。このファクターは、コンテキスト diff リストの内容の行数 (通常は 3) よりも大きい値には設定できません。
- 注: 大きい値の fuzz ファクターを使用すると、正確でないパッチの可能性が高くなります。
- i PatchFile** パッチ情報を、標準入力からではなく、指定されたファイルから読みます。
- l** (小文字の L) diff リスト・スクリプト内のブランク文字のシーケンスを、入力ファイル内のブランク文字のシーケンスと一致させます。その他の文字は正確に突き合わされます。
- n** スクリプトを通常の diff リストとして解釈します。このフラグは、**-c** または **-e** フラグと一緒に指定できません。
- N** 既にファイルに差分が適用されているパッチを無視します。デフォルトでは、既に差分が適用されているパッチはリジェクトされます。
- o OutFile** パッチされるファイルをコピーし、変更を適用し、修正されたバージョンを指定した出力ファイルに書きます。単一ファイルへの複数のパッチは、以前のパッチで作成されたファイルの中間バージョンに適用されます。したがって、複数のパッチを適用すると、出力ファイルの複数のバージョンが連結されます。

<b>-p</b> <i>Number</i>	<p>パス名ストリップ・カウントを設定します。このカウントは、パッチ・ファイル内で見つかったパス名の処理方法を制御します。このフラグは、指定したパスとは異なるディレクトリ内にファイルを保持する場合に便利です。ストリップ・カウントは、パス名の前からストリップされるスラッシュの数を指定します。中間のディレクトリ名もストリップされます。例えば、パッチ・ファイルで <code>/u/leon/src/blurf1/blurf1.c</code> を指定したと仮定すると、</p> <ul style="list-style-type: none"> <li>• <b>-p 0</b> を指定すると、パス名全体が変更されないままになります。</li> <li>• <b>-p 1</b> を指定すると、先行スラッシュが除去され、<code>u/leon/src/blurf1/blurf1.c</code> となります。</li> <li>• <b>-p 4</b> を指定すると、4 つのスラッシュと 3 つのディレクトリが除去され、<code>blurf1/blurf1.c</code> となります。</li> </ul> <p><b>-p</b> フラグを指定しないと、ベース名 (パス名の最後のコンポーネント) だけが使用されます。このフラグは、<i>File</i> パラメーターが指定されない場合にのみ働きます。</p>
<b>-r</b> <i>RejectFile</i>	<p>デフォルトのリジェクト・ファイル名を上書きします。デフォルトのリジェクト・ファイル名は、元のファイル名に接尾部 <code>.rej</code> を付加して作成されます。</p>
<b>-R</b>	<p>パッチ・スクリプトの意味を反転させます。例えば、diff リストを新規バージョンから旧バージョンに対して作成する場合、<b>-R</b> フラグを指定すると、<b>patch</b> コマンドは、スクリプトの各部分を適用する前に反転させます。リジェクトされた差分は、スワップ後のフォーマットで保存されます。<b>-R</b> フラグは <code>ed</code> スクリプトと一緒に使用できません。これは、反転操作を再構成するための情報が少な過ぎるからです。<b>-R</b> フラグを指定しないと、<b>patch</b> コマンドは、パッチ・ファイルの一部が正常に適用されるまで、各部分を通常の意味でも逆の意味でも適用しようとします。この試みが正常に終了すると、<b>-R</b> フラグを設定するかどうか判断を求めるプロンプトが表示されます。</p> <p><b>注:</b> この方法は、最初のコマンドが追加である (つまり、反転で削除になっていた) 通常 diff リストで使用する場合には、反転されたパッチを検出できません。null のコンテキストはどこにでも一致するので、追加は常に成功します。幸い、ほとんどのパッチでは行を削除するのではなく、行を追加または変更します。したがって、反転された通常の diff リストは、ほとんど削除で始まるので失敗し、試行錯誤を引き起こします。</p>
<b>-s</b>	<p>エラーが発生しない限り、何も表示せずにパッチします。</p>
<b>-v</b>	<p>改訂ヘッダーとパッチ・レベルを出力します。<b>-v</b> フラグをほかのフラグと一緒に指定すると、ほかのフラグは無視されます。</p>
<b>-x</b> <i>Number</i>	<p>内部デバッグ・フラグを設定します。このフラグは <b>patch</b> コマンド開発者専用です。</p>

## 終了状況

以下の終了値が戻されます。

- 0 正常終了。
- 1 エラーが発生しました。

## 例

1. `difflisting` ファイル内の diff リストをファイル `prog.c` に適用するには、以下のように入力します。

```
patch -i difflisting prog.c
```
2. `prog.c` ファイルの元のバージョンを保存するには、以下のように入力します。

```
patch -b -i difflisting prog.c
```

これにより、変更が `prog.c` に適用され、`prog.c` の元の内容はファイル `prog.c.orig` に保存されます。

- 元のバージョンを変更せずに `prog.c` ファイルをパッチするには、以下のように入力します。

```
patch -i difflisting -o prog.new prog.c
```

これにより、`prog.c` がソース・ファイルとして使用されますが、変更後のバージョンは `prog.new` というファイルに書かれます。

## ファイル

`/usr/bin/patch` `patch` コマンドが入っています。

## 関連情報

`diff` コマンド、`ed` コマンド。

---

## pathchk コマンド

### 目的

パス名を検査します。

### 構文

```
pathchk [ -p ] PathName...
```

### 説明

`pathchk` コマンドは、1 つまたは複数のパス名が、有効で移植可能であるかどうかを検査します。デフォルトでは、`pathchk` コマンドは基本のファイルシステムに基づいて、`path name` パラメーターで指定された、それぞれのパス名のそれぞれのコンポーネントを検査します。以下の条件に該当するパス名の場合はエラー・メッセージが送信されます。

- 絶対パス名の長さ (バイト) が、システムで許容される長さを超える場合。
- コンポーネントの長さ (バイト) が、システムで許容される長さを超える場合。
- コンポーネントに検索が許可されていない場合。
- コンポーネントの文字がその下位のディレクトリー内で無効である場合。

`PathName` パラメーターで指定したパス名と一致するファイルが作成でき、そのファイルが上に述べた条件に当てはまらない場合は、パス名の 1 つまたは複数のコンポーネントが存在しなくてもエラーとは見なされません。

`-p` フラグを指定すると、さらに広範な移植性検査ができます。

## フラグ

- p** POSIX 移植性標準に基づいてパス名を検査します。以下の条件に該当するパス名の場合はエラー・メッセージが送信されます。
- 絶対パス名の長さ (バイト) が POSIX 標準で許容される長さを超える場合。
  - コンポーネントの長さ (バイト) が POSIX 標準で許容される長さを超える場合。
  - コンポーネントの文字が移植可能なファイル名文字セットにない場合。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0** すべての *PathName* オペランドがすべての検査にパスしました。  
**>0** エラーが発生しました。

## 例

- システムでの `/home/bob/work/tempfiles` パス名の有効性と移植性を検査するには、以下のように入力します。

```
pathchk /home/bob/work/tempfiles
```

- POSIX 標準に対する `/home/bob/temp` パス名の有効性と移植性を検査するには、以下のように入力します。

```
pathchk -p /home/bob/temp
```

## ファイル

`/usr/bin/pathchk` **pathchk** コマンドが入っています。

## 関連情報

**mhpath** コマンド。

オペレーティング・システムおよびデバイスのマネージのファイルシステムのセクション。

---

## pax コマンド

### 目的

アーカイブ・ファイルのメンバーを取り出して書き込み、リストします。ファイルおよびディレクトリー階層をコピーします。

### 構文

#### アーカイブ・ファイルのメンバー・ファイルをリストする

```
pax [ -c ] [ -d ] [ -n ] [ -U ] [ -v ] [ -H | -L ] [ -f Archive ] [ -s ReplacementString ... ]  
[ -x Format ] [ -o Options ] [ Pattern ... ]
```

## **-r** フラグを使用してアーカイブ・ファイルを取り出す

```
pax -r [ -c ] [ -d ] [ -i ] [ -k ] [ -n ] [ -U ] [ -u ] [ -v ] [ -H | -L ] [ -f Archive ] [ -o Options ] [ -p String ... ] [ -s ReplacementString ... ] [ -x Format ] [ Pattern ... ]
```

## **-w** フラグを使用してアーカイブ・ファイルに書き込む

```
pax -w [ -d ] [ -i ] [ -t ] [ -U ] [ -u ] [ -v ] [ -X ] [ -H | -L ] [ -E ] [ -b Blocking ] [ -a ] [ -f Archive ] [ -o Options ] [ -s ReplacementString ... ] [ -x Format ] [ File ... ]
```

## **-r** フラグと **-w** フラグを使用してファイルをコピーする

```
pax -r -w [ -d ] [ -i ] [ -k ] [ -l ] [ -t ] [ -U ] [ -u ] [ -v ] [ -X ] [ -H | -L ] [ -p String ... ] [ -o Options ] [ -s ReplacementString ... ] [ -x Format ] [ File ... ] Directory
```

## 説明

**pax** コマンドは、アーカイブ・ファイルのメンバー・ファイルを取り出して書き込みます。アーカイブ・ファイルのメンバー・ファイルのリストを書き込み、ディレクトリーの階層をコピーします。**-r** フラグと **-w** フラグはアーカイブ操作のタイプを指定します。

注: **pax** は、復元されているファイルをアクティブに分散させます。ファイルに、NULL が記入されていて、ブロック調整およびサイズ変更された区域がある場合、**pax** によって、それらのファイルシステム・ブロック用の物理スペースは割り当てられません。バイト単位でのファイルのサイズは変わりませんが、ファイルシステム内で実際に使われるスペースは、NULL 以外の区域専用になります。

## アーカイブ・ファイルのメンバー・ファイルのリスト (リスト・モード)

**-r** フラグも **-w** フラグも指定されない場合、**pax** コマンドは、標準入力から読んだアーカイブ・ファイルのすべてのメンバー・ファイルをリストします。*Pattern* パラメーターを指定すると、指定したパターンに一致するパス名を持つメンバー・ファイルだけが標準出力に書き出されます。指定されたファイルがディレクトリーの場合は、そのディレクトリーに含まれているファイル階層も取り出されます。**-r** フラグも **-w** フラグも指定しない場合は、**-c**、**-d**、**-f**、**-n**、**-s**、**-v** フラグ、および *Pattern* パラメーターを指定できます。

## **-r** フラグを使用したアーカイブ・ファイルの取り出し (読み取りモード)

**-r** フラグは指定するが、**-w** フラグは指定しない場合、**pax** コマンドは標準入力から読んだアーカイブ・ファイルのすべてのメンバー・ファイルを取り出します。*Pattern* パラメーターを指定すると、指定したパターンに一致するパス名を持つメンバー・ファイルだけが標準出力に書き出されます。指定されたファイルがディレクトリーの場合は、そのディレクトリーに含まれているファイル階層も取り出されます。**-r** フラグは、**-c**、**-d**、**-f**、**-i**、**-k**、**-n**、**-s**、**-u**、**-v** フラグ、および *Pattern* パラメーターと一緒に指定できます。

取り出したファイルのアクセス時間および修正時間は、アーカイブ・ファイルの場合と同じです。取り出したファイルのファイル・モードは、ファイルをアーカイブした場合と同じですが、ユーザーのデフォルトのファイル作成モード (**umask**) の影響を受けない場合に限られます。取り出したファイルの **S\_ISUID** ビットと **S\_ISGID** ビットはクリアされます。

アーカイブ・メンバーを取り出すのに中間ディレクトリーが必要な場合には、**pax** コマンドは、**S\_IRWXU**、**S\_IRWXG**、**S\_IRWXO** のマスクの値のビット単位の包含 OR として設定されたアクセス権を持つディレクトリーを作成します。

選択したアーカイブ・フォーマットがリンクされるファイルの指定をサポートする場合、そのアーカイブ・フォーマットが取り出されたときにそれらのファイルをリンクできないとエラーになります。

## **-w** フラグを使用したアーカイブ・ファイルの書き込み (書き込みモード)

**-w** フラグを指定して **-r** フラグを指定しない場合、**pax** コマンドは、*File* パラメーターで指定されたファイルの内容をアーカイブ・フォーマットで標準出力に書き出します。*File* パラメーターを指定しないと、コピーするファイルのリストが標準入力から 1 行に 1 つずつ読み取られます。*File* パラメーターがディレクトリーを指定する場合は、そのディレクトリーに含まれるすべてのファイルが書き込まれます。**-w** フラグは、**-a**、**-b**、**-d**、**-f**、**-i**、**-o**、**-s**、**-t**、**-u**、**-v**、**-x**、**-X** フラグ、および *File* パラメーターと一緒に指定できます。

United States Tape Archiver (USTAR) のフォーマットに 100 文字を超えるパス名を指定する場合は、パス名が接頭部バッファー、/ (スラッシュ) 1 つ、およびネーム・バッファーで構成されることに注意してください。接頭部バッファーには最大 155 文字まで入り、ネーム・バッファーには最大 100 文字まで入りません。パス名をスラッシュでこの 2 つの部分に分割できなければ、アーカイブできません。この制限は **tar** ファイルを圧縮するアーカイブ・ヘッダーの構造によるものであり、規格に準拠し下位互換性を保つために守らなくてはなりません。

## **-r** フラグと **-w** フラグを使用したファイルのコピー (コピー・モード)

**-r** フラグと **-w** フラグをともに指定する場合、**pax** コマンドは、*File* パラメーターで指定されたファイルを *Directory* パラメーターで指定された出力先ディレクトリーにコピーします。ファイルを指定しないと、コピーするファイルのリストが、1 行に 1 つずつ標準入力から読み取られます。指定ファイルがディレクトリーである場合には、そのディレクトリーに含まれているファイル階層もコピーされます。**-r** フラグと **-w** フラグは、**-d**、**-i**、**-k**、**-l**、**-o**、**-p**、**-s**、**-t**、**-u**、**-v**、**-X** フラグ、および *File* パラメーターと一緒に指定できます。*Directory* パラメーターは必ず指定してください。

コピーしたファイルは、アーカイブ・ファイルへ書き込み、その後に取り出した場合と同じですが、元のファイルとコピーしたファイルの間にハード・リンクがある可能性の点が異なります。

## **-o** フラグを使用したアーカイブ・アルゴリズムの修正

**-o** フラグを使用して、キーワードと値の組に従ってアーカイブ・アルゴリズムを修正します。キーワードと値の組は、正しいアーカイブ・フォーマットに準拠している必要があります。有効なキーワードとその動作のリストは、後で、**-o** フラグの説明のところで出てきます。

## その他の注意事項

読み取りまたはコピー・モードでは、中間ディレクトリーがアーカイブ・メンバーを取り出す必要がある場合、**pax** は、パス引数として中間ディレクトリーを使用し、モード引数として値 **S\_IRWXU** を使用する **mkdir()** サブルーチンと同じアクションを取ります。

指定されたパターンまたはファイル・オペランドが少なくとも 1 つのファイルまたはアーカイブ・メンバーと一致しない場合、**pax** は、一致しなかったそれぞれのファイルまたはアーカイブ・メンバーごとに診断メッセージ 1 つずつを標準エラーに書き込み、エラー状況を表して終了します。

スキャン・ディレクトリーでは、**pax** は無限ループ (例えば、最後に訪問されたファイルの祖先であり、既に訪問されているディレクトリーを入力するなど) を検出します。無限ループを検出すると、**pax** は診断メッセージを標準エラーに書き込み、終了します。

**pax** が、**-x pax** アーカイブ・フォーマットを使用した読み取りモードまたはリスト・モードのときに、拡張ヘッダー・レコード内のファイル名、リンク名、オーナー名などのフィールドがすべて **pax** UTF8 コード集合フォーマットから現行コード集合およびロケールに変換できない場合、**pax** は診断メッセージを標準エラーに書き込み、**-o invalid=** オプションの場合の説明に従ってそのファイルを処理してから、アーカイブ内の次のファイルを処理します。



AIX 5.3 の場合、**pax** コマンドは、デフォルトでは拡張属性を無視します。 **-U** オプションは **pax** に、ACL を含む拡張属性をアーカイブまたは復元することを通知します。 **-pe** オプションは ACL を保存します。 **-pe** オプションが指定されているときに **pax** が ACL を保存するのに失敗すると、標準エラーに診断メッセージが書き込まれますが、抽出されたファイルは削除されません。ゼロ以外の終了コードが戻されません。 **pax** アーカイブ・ファイル内に、拡張属性用に新しいレコード・タイプが必要です。

## 変数

*Directory*

ファイルをコピーするとき、出力先ディレクトリーのパスを指定します。

*File*

コピーまたはアーカイブするファイルのパスを指定します。 *File* パラメーターと一致するファイルがない場合、**pax** コマンドは、エラーを検出して終了し、診断メッセージを書きます。

*Pattern*

アーカイブ・メンバーの 1 つまたは複数のパスに一致するパターンを指定します。 \  
(バックスラッシュ) は、*Pattern* パラメーターでは識別されず、このためにその後の文字が意味を持たなくなります。 *Pattern* パラメーターを指定しない場合、すべてのメンバーがアーカイブで選択されます。

*Pattern* パラメーターが指定されているが、指定されたパターンに一致するアーカイブ・メンバーが見つからない場合、**pax** コマンドは、エラーを検出して終了し、診断メッセージを書きます。

## フラグ

**-a**

アーカイブの最後にファイルを追加します。

**注:** ストリーミング・テープ・デバイスの場合、追加はできません。

**-b** *Blocking*

出力用のブロックのサイズを指定します。 *Blocking* パラメーターは正の 10 進整数値を指定し、この値がブロックごとのバイト数を指定します。 POSIX2 に準拠するアプリケーションでは、32256 を超えるブロック・サイズを指定しないでください。デバイスおよびアーカイブ・フォーマットは、ブロック化に制限を付けることがあります。ブロック化は、入力時に自動的に決定されます。アーカイブを作成する場合のデフォルト・ブロック化は、アーカイブ・フォーマットによって左右されます。 (**-x** フラグの定義を参照。)

*Blocking* パラメーターの値は、以下のいずれかになります。

*Integer* **b**

バイト単位のブロック・サイズが、*Integer* パラメーターの値に 512 を掛けた値を指定した正の 10 進整数になるように指定します。

*Integer* **k**

バイト単位のブロック・サイズが、*Integer* パラメーターの値に 1024 を掛けた値を指定した正の 10 進整数になるように指定します。

*Integer* **m**

バイト単位のブロック・サイズが、*Integer* パラメーターの値に 1024 x 1024 を掛けた値を指定した正の 10 進整数になるように指定します。

*Integer+Integer*

バイト単位のブロック・サイズが、*Integer* パラメーターで指定した正の 10 進整数の合計になるように指定します。

**-c**

*Pattern* パラメーターで指定したもの以外の、すべてのファイルまたはアーカイブ・メンバーに一致します。



- d** コピー、アーカイブ、または取り出されるディレクトリーが、ディレクトリーの内容ではなく、ディレクトリーそのものだけに一致するようにします。
- E** 新規のアーカイブ/既存のアーカイブにファイルが追加される際に、長いユーザー名およびグループ名が切り捨てられないようにします。
- f Archive** 標準入力 (**-w** フラグを指定しない場合) または標準出力 (**-w** フラグは指定するが **-r** フラグは指定しない場合) の代わりに使用するアーカイブ・ファイルのパスを指定します。**-a** フラグのオプションと一緒に指定すると、アーカイブに書き込まれるすべてのファイルがアーカイブの最後に追加されます。
- H** ディレクトリーを参照するシンボリック・リンクがコマンド行に指定されている場合、**pax** は、リンクの名前をファイル階層の名前として使用して、そのリンク内で参照されているディレクトリー内にルートがあるファイル階層をアーカイブします。デフォルトでは、**pax** は、シンボリック・リンクそのものをアーカイブします。
- i** ファイルまたはアーカイブの名前を対話型で変更します。*Pattern* パラメーターに一致するそれぞれのアーカイブ・メンバー、または *File* パラメーターに一致するファイルについては、ファイルまたはアーカイブ・メンバーの名前を含むディスプレイ・デバイスにプロンプトが表示されます。そこで、1 行がディスプレイ・デバイスから読み取られます。この行が空である場合には、ファイルまたはアーカイブ・メンバーはスキップされます。この行が単一のピリオドで構成されている場合、その名前を修正せずにファイルまたはアーカイブ・メンバーが処理されます。そうでない場合には、その名前がその行の内容に置換されます。
- k** **pax** コマンドが既存ファイルに書き込むのを防ぎます。
- l** ファイルをコピーするときに、ファイルをリンクします。可能な場合にはいつでも、ソース・ファイルと出力先ファイルの階層間にハード・リンクが設けられます。
- L** ディレクトリーを参照するシンボリック・リンクが、コマンド行に指定されたか、ファイル階層のスキャン中に検出された場合、**pax** は、リンクの名前をファイル階層の名前として使用して、そのリンク内で参照されているディレクトリー内にルートがあるファイル階層をアーカイブします。デフォルトでは、**pax** は、シンボリック・リンクそのものをアーカイブします。
- n** それぞれの *Pattern* パラメーターに一致する最初のアーカイブ・メンバーを選択します。それぞれのパターンについて、複数のアーカイブ・メンバーが一致することはありません。

## -o Options

*Options* パラメーターとして指定したキーワードと値の組に従って、アーカイブ・アルゴリズムを変更します。キーワードと値の組は以下のフォーマットでなくてはなりません。

*keyword:=value,keyword:=value,...*

キーワードによっては、それぞれの説明で示されているように、特定のファイル・フォーマットにだけ適用されるものがあります。処理されているファイル・フォーマットに適用できないキーワードを使用しても、**pax** によって無視されます。

キーワードの前にはホワイト・スペースを設けることができます。 *value* フィールドは、ゼロ桁以上の文字で構成されます。 *value* の中では、リテラル・コンマの前にはバックスラッシュ (\) を付ける必要があります。最後の文字としてのコンマや、最後の文字としてホワイト・スペースだけが後に付いているコンマは、 *Options* 内では無視されます。 **-o** オプションは複数指定できます。これらの複数の **-o** オプションに指定されたキーワードが競合する場合、コマンド行シーケンスの中で後の方に出てくるキーワードと値が優先されます。先に出てきた値は無視されます。

次のキーワードと値の組は、示されたファイル形式の場合にサポートされます。

**datastream=pathname,datastr\_size=size** (すべてのファイル形式に適用されます。)

**datastream** キーワードは、着信アーカイブ・ファイルがファイル・フォーマットではなく、標準入力デバイスからのデータ・ストリームであることを示します。したがって、データは **-x** フラグで認識されるフォーマットの正規ファイルとしてアーカイブしてはなりません。データ・ストリームのファイル名は、 *pathname* パラメーターで指定しなくてはならず、コマンドを起動した人の ID、グループ ID、ファイル・モードの **umask** を組み込まなくてはなりません。

**注:** **datastream** キーワードには、デフォルトの変数サイズがありません。変数サイズを指定しなくてはなりません。

**datastr\_size** キーワードは、データ・ストリーム入力のサイズをバイト単位の 10 進数を使って示します。 **pax** コマンドは、 *size* パラメーターを読む前にファイル終わり (EOF) に達すると、アーカイブ・ファイルを null 値で埋めます。 null 値の場合、アーカイブ・ファイルのサイズは *size* パラメーターで指定したものと同じになります。アーカイブ・ファイル内のデータが指定したサイズを超えると、 **pax** コマンドは *size* パラメーターで指定されたサイズになるようにアーカイブ・ファイルを切り捨てます。また、 **pax** コマンドは入力の取り出しを停止してアーカイブ・ファイルをクローズします。

**注:** キーワードの組については複数のインスタンスを指定できます。同じキーワードに異なる値を割り当てると、 **pax** コマンドはそのキーワードに最後に割り当てた値を使用して **-o** フラグを実行します。

**delete=pattern (-x pax** フォーマットのものに適用されます。)

**pax** は、書き込みモードまたはコピー・モードで使用される場合、それが作成する拡張ヘッダー・レコードからの *pattern* とマッチングするキーワードをすべて省略します。読み取りモードまたはリスト・モードで使用される場合、 **pax** は、その拡張ヘッダー・レコード内の *pattern* とマッチングするキーワードをすべて無視します。いずれの場合も、マッチングは、標準シェル・パターン・マッチング表記を使用して行われます。例えば、 **-o delete=security.\*** は、セキュリティ関連情報を抑止します。

## -o Options (続き)

**exthdr.name=string (-x pax** フォーマットのみに適用されます。)

このキーワードによって、ユーザーが、拡張ヘッダー・レコードの **ustar** ヘッダー・ブロック内に書き込まれる名前を制御できます。この名前は、次の文字置換が行われた後の *string* の内容です。

string に組み込む内容:

置換後の内容:

**%d** ファイルのディレクトリー名。変換されたパス名上の **dirname** ユーティリティーの結果と同じ。

**%f** ファイルのファイル名。変換されたパス名上の **basename** ユーティリティーの結果と同じ。

**%%** %% 文字

*string* 内で上記以外の % 文字を使用すると、未定義の結果が生じます。このキーワードと値の組が **-o Options** リストで指定されていない場合、その名前のデフォルト値は次のようになります。

%d/PaxHeaders/%f

**globexthdr.name=string (-x pax** フォーマットのみに適用されます。)

**pax** は、該当のオプションを指定した書き込みモードまたはコピー・モードで使用されると、以前のバージョンの **pax** によって正規ファイルとして扱われる、**ustar** ヘッダー・ブロックを含んだグローバル拡張ヘッダー・レコードを作成します。このキーワードによって、ユーザーが、グローバル拡張ヘッダー・レコードの **ustar** ヘッダー・ブロック内に書き込まれる名前を制御できます。この名前は、次の文字置換が行われた後の *string* の内容です。

string に組み込む内容:

置換後の内容:

**%n** 1 で始まる、アーカイブ内のグローバル拡張ヘッダー・レコードのシーケンス番号を表す整数。

**%%** % 文字

*string* 内で上記以外の % 文字を使用すると、未定義の結果が生じます。このキーワードと値の組が **-o Options** リストで指定されていない場合、その名前のデフォルト値は次のようになります。

\$TMPDIR/GlobalHead.%n

ここで、**\$TMPDIR** は **TMPDIR** 環境変数の値、または **TMPDIR** が設定されていない場合は **/tmp** です。

**invalid=action (-x pax** フォーマットのみに適用されます。)

このキーワードによって、ユーザーが、次の状態の拡張ヘッダー・レコード内で値を検出したときに **pax** が取るアクションを制御できます。

- 読み取りモードまたはコピー・コードの場合に、宛先階層内で無効である。
- リスト・モードの場合に、コード集合と現行ロケール内に書き込むことができない。

## -o Options (続き)

**pax** は次の無効値を認識します。

- 読み取りモードまたはコピー・モードの場合に、宛先階層で無効な文字エンコードを含むファイル名またはリンク名。(例えば、この名前には組み込み NULL が含まれている場合があります。)
- 読み取りモードまたはコピー・モードの場合に、出力先階層で可能な最大長を超えるファイル名またはリンク名 (パス名のコンポーネントか、パス名全体のどちらかの場合)。
- リスト・モードの場合に、コード集合および現行ロケールに書き込むことができない文字列値 (ファイル名、リンク名、ユーザー名など)。

次に示す、*action* 引数の相互排他的な値がサポートされています。

### • **bypass**

読み取りモードまたはコピー・モードの場合、**pax** はファイルをバイパスして、それにより出力先階層への変更は行われません。リスト・モードの場合、**pax** は、そのファイルについて要求された有効値をすべて書き込みますが、無効値を書き込むためのメソッドは指定されていません。

### • **rename**

読み取りモードまたはコピー・モードの場合、**pax** は、無効なファイル名またはリンク名を持つ各ファイルごとに **-i** フラグが有効であるものとした働きをして、ユーザーが対話式に置換名を指定できるようにします。リスト・モードでは、**pax** は **bypass** アクションとまったく同じ動作をします。

### • **UTF8**

読み取り、コピー、またはリスト・モードで使用される場合に、拡張ヘッダー・レコード内のファイル名、リンク名、オーナー名などのフィールドがすべて、**pax UTF8** コード集合形式から現行コード集合およびロケールに変換できない場合、**pax** はその名前に実際の UTF8 エンコードを使用します。

### • **write**

読み取りモードまたはコピー・モードの場合、**pax** は、ファイルを書き込んで、名前の変換または切り捨てを行います。これは、有効名を持つ既存ファイルが上書きされるかどうかに関係なく行われます。リスト・モードでは、**pax** は **bypass** アクションとまったく同じ動作をします。

**-o invalid=action** が指定されない場合、**pax** は、**bypass** アクションが指定されたときと同様の働きをします。**-o invalid=actions** で認められている既存ファイルの上書きは、許可 (**-p**) と変更時間 (**-u**) の制限の対象となり、**-k** フラグも指定されている場合は抑止されます。

**linkdata (-x pax** フォーマットのものに適用されます。)

書き込みモードでは、**pax** は、ファイルが単に、アーカイブに既に内容が書き込まれているファイルに対するハード・リンクであっても、そのアーカイブにファイルの内容を書き込みます。

**-o Options** (続き)

**listopt=format** (すべてのファイル形式に適用されます。)

このキーワードは、**-v** オプションがリスト・モードで指定されているときに生成される目次の出力形式を指定します。このキーワードと値の組は、混同されないように、**-o** フラグ以降の唯一かつ最後のキーワードと値の組として使用しなければなりません。オプションと引数の組の残りの部分にある文字はすべて、フォーマット文字列の一部と見なされます。複数の **-o listopt=format** オプションを指定すると、フォーマット文字列は 1 つの連結された文字列と見なされ、コマンド行の順序で評価されます。詳細は、**リスト・モード・フォーマットの仕様**のセクションを参照してください。

**times (-x pax** フォーマットのみ適用されます。)

**pax** は、書き込みモードまたはコピー・モードで使用される場合、各ファイルごとに **atime**、**ctime**、および **mtime** 拡張ヘッダー・レコードを組み込みます。

## 拡張ヘッダー・キーワード

**(-x pax** フォーマットのみ適用されます。)

**-x pax** フォーマットを指定すると、下記のリストで定義されたキーワードと値はすべて、次の 2 つのモードのどちらかで **-o** フラグへのパラメーターとして使用できます。

*keyword=value*

書き込みモードまたはコピー・モードで使用される場合、これらのキーワードと値の組は、新しいアーカイブのグローバル拡張ヘッダー・レコードに書き込まれます。読み取りモードまたはリスト・モードで使用される場合、これらのキーワードと値の組は、読み取られるアーカイブのグローバル拡張ヘッダー・レコード内にそれらの組がある場合と同様の働きをします。どちらの場合も、その値は、指定されたキーワードの個々の拡張ヘッダー・レコード内で値が割り当てられていない、すべてのファイルに適用されます。

*keyword:=value*

書き込みモードまたはコピー・モードで使用される場合、これらのキーワードと値の組は、新しいアーカイブの各ファイルの拡張ヘッダー・レコードに書き込まれます。読み取りモードまたはリスト・モードで使用される場合、これらのキーワードと値の組は、読み取られるアーカイブの各ファイルの拡張ヘッダー・レコード内にそれらの組がある場合と同様の働きをします。どちらの場合も、その値は、グローバルまたはファイル特定の拡張ヘッダー・レコードにある指定のキーワードの値をすべてオーバーライドします。

### atime

後に続くファイル (1 つまたは複数) のファイル・アクセス時間。ファイルの **stat** 構造体の **st\_atime** メンバーの値と同じ。

### charset

後に続くファイル (1 つまたは複数) のデータをエンコードするために使用する文字集合の名前。このテーブルのエントリは、既知の標準を参照するために定義されます。

値

"ISO-IR 646 1990"

"ISO-IR 8859 1 1987"

標準

ISO/IEC 646 IRV

ISO 8859-1

"ISO-IR 8859 2 1987"  
"ISO-IR 10646 1993"  
"ISO-IR 10646 1993 UTF8"  
"BINARY"

ISO 8859-2  
ISO/IEC 10646  
ISO/IEC 10646, UTF8 encoding  
なし

エンコードは、通知だけを目的として拡張ヘッダーに組み込まれています。 **pax** は、説明に従って使用すると、ファイル・データを他のどのエンコードにも変換しません。 **BINARY** エントリーは、エンコードされていないバイナリー・データを示します。

### comment

コメントとして使用する一連の文字。 **pax** は値フィールド内の文字をすべて無視します。

### ctime

以下のファイルのファイル作成時間。ファイルの `stat` 構造体の `st_ctime` メンバーの値と同じ。

### gid

ファイルを所有するグループのグループ ID。10 進数で表わす。このレコードは、後に続くヘッダー・ブロック (1 つまたは複数) 内の `gid` フィールドをオーバーライドします。 **pax** は、書き込みモードまたはコピー・モードで使用されると、グループ ID が 99,999,999 を超えるファイルごとに、`gid` 拡張ヘッダー・レコードを 1 つずつ組み込みます。

### gname

グループ・データベース内でグループ名としてフォーマットされている、後に続くファイル (1 つまたは複数) のグループ。このレコードは、後に続くヘッダー・ブロックの `gid` フィールドと `gname` フィールド、および `gid` 拡張ヘッダー・レコードをオーバーライドします。読み取り、コピー、またはリスト・モードで使用されると、**pax** は、ヘッダー・レコードの UTF8 エンコードからの名前を、受信システム上のグループ・データベースに適した文字集合に変換します。いずれかの UTF8 文字が変換できない場合に、**-o invalid=UTF8** オプションが指定されていないと、未定義の結果が生じます。 **pax** は、書き込みモードまたはコピー・モードで使用されると、グループ名がポータブル文字集合の文字と数字だけで表すことができないファイルごとに、`gname` 拡張ヘッダー・レコードを 1 つずつ組み込みます。

### linkpath

以前にアーカイブされている別のファイル (任意のタイプ) に向けて作成中のリンクのパス名。このレコードは、後に続く **ustar** ヘッダー・ブロック (1 つまたは複数) の `linkname` フィールドをオーバーライドします。

後に続く **ustar** ヘッダー・ブロックが、作成されるリンクのタイプ (ハードまたはシンボリック) を決定します。後者の場合、リンク・パス値が、そのシンボリック・リンクの内容になります。 **pax** は、リンクの名前 (シンボリック・リンクの内容) を、UTF8 エンコードからローカル・ファイルシステムに適した文字集合に変換します。

**pax** は、書き込みモードまたはコピー・モードで使用されると、パス名を `NULL` 以外のポータブル文字集合のメンバーだけで表すことができないリンクごとに `linkpath` 値を 1 つずつ組み込みます。

### mtime



後に続くファイル (1 つまたは複数) のファイル修正時間。ファイルの `stat` 構造体の `st_mtime` メンバーの値と同じ。このレコードは、後に続くヘッダー・ブロック (1 つまたは複数) の `mtime` フィールドをオーバーライドします。変更時間は、そのプロセスに、復元するために適切な特権があれば復元されます。

## path

後に続くファイル (1 つまたは複数) のパス名。このレコードは、後に続くヘッダー・ブロック (1 つまたは複数) の `name` フィールドと `prefix` フィールドをオーバーライドします。**pax** は、ファイルのパス名を、UTF8 エンコードからローカル・ファイルシステムに適した文字集合に変換します。**pax** は、書き込みモードまたはコピー・モードで使用されると、パス名を `NULL` 以外のポータブル文字集合のメンバーだけで表すことができないファイルごとに、`path` 拡張ヘッダー・レコードを 1 つずつ組み込みます。

## realtime.any

`realtime` が接頭部として付いたキーワードは、将来の POSIX リアルタイム標準化のために予約されます。**pax** はそれらを認識しますが、何もせずに無視します。

## security.any

`security` が接頭部として付いたキーワードは、将来の POSIX セキュリティ標準化のために予約されます。**pax** はそれらを認識しますが、何もせずに無視します。

## size

オクテット単位でのファイルのサイズ。ISO/IEC 646 からの数字を使用する 10 進数で表されます。このレコードは、後に続くヘッダー・ブロック (1 つまたは複数) の `size` フィールドをオーバーライドします。**pax** は、書き込みモードまたはコピー・モードで使用されると、サイズ値が 999,999,999,999 を超えるファイルごとに、拡張ヘッダー・レコードのサイズを組み込みます。

## uid

ファイルを所有するユーザーのユーザー ID。ISO/IEC 646 からの数字を使用する 10 進数で表されます。このレコードは、後に続くヘッダー・ブロック (1 つまたは複数) の `uid` フィールドをオーバーライドします。**pax** は、書き込みモードまたはコピー・モードで使用されると、オーナー ID が 99,999,999 を超えるファイルごとに、`uid` 拡張ヘッダー・レコードを 1 つずつ組み込みます。

## uname

ユーザー・データベース内のユーザー名としてフォーマットされている、後に続くファイル (1 つまたは複数) のオーナー。このレコードは、後に続くヘッダー・ブロックの `uid` フィールドと `uname` フィールド、および `uid` 拡張ヘッダー・レコードをオーバーライドします。読み取り、コピー、またはリスト・モードで使用されると、**pax** は、ヘッダー・レコード内の UTF8 エンコードからの名前を、受信システム上のデータベースに適した文字集合に変換します。いずれかの UTF8 文字が変換できない場合に、**-o invalid=UTF8** オプションが指定されていないと、未定義の結果が生じます。**pax** は、書き込みモードまたはコピー・モードで使用されると、ユーザー名がポータブル文字集合の文字と数字だけで表すことができないファイルごとに、`uname` 拡張ヘッダー・レコードを 1 つずつ組み込みます。

`value` フィールドの長さがゼロの場合、ヘッダー・ブロック・フィールド、以前に入力された拡張ヘッダー値、または同じ名前のグローバル拡張ヘッダー値のどれかが削除されます。



拡張ヘッダー・レコード (または **-o** オプションと引数の組の中) のキーワードが、**ustar** ヘッダー・ブロック内の対応するフィールドをオーバーライドまたは削除すると、**pax** はそのヘッダー・ブロック・フィールドの内容を無視します。

## 拡張ヘッダー・キーワードの優先度

(**-x pax** フォーマットのみ適用されます。)

このセクションでは、さまざまなヘッダー・レコードとファイル、およびコマンド行オプションを、アーカイブ内のファイルに適用するように選択する場合の優先度について説明します。**pax** は、読み取りモードまたはリスト・モードで使用すると、次のシーケンスでファイル属性を判別します。

1. **-o delete=keyword-prefix** が使用される場合、影響を受ける属性は、ステップ (7) が適用可能であればこのステップから判別されます。使用されない場合は無視されます。
2. **-o keyword:=NULL** が使用される場合、影響を受ける属性は無視されます。
3. **-o keyword:=value** が使用される場合、影響を受ける属性には値が割り当てられます。
4. ファイル特定の拡張ヘッダー・レコードに *value* が存在する場合、影響を受ける属性には値が割り当てられます。拡張ヘッダー・レコード同士が競合する場合、ヘッダー内で最後に指定されたものが優先されます。
5. **-o keyword=value** が使用される場合、影響を受ける属性には値が割り当てられます。
6. グローバル拡張ヘッダー・レコードに値が存在する場合、影響を受ける属性には値が割り当てられません。グローバル拡張ヘッダー・レコード同士が競合する場合、グローバル・ヘッダー内で最後に指定されたものが優先されます。
7. 上記以外の場合、属性は、**ustar** ヘッダー・ブロックから判別されます。

### **-p** *String*

取り出すときに保存するかまたは破棄する 1 つまたは複数のファイル特性を指定します。*String* パラメーターは、文字 **a**、**e**、**m**、**o**、**p** で構成されます。同一文字列内で複数の特性を組み合わせることができ、複数の **-p** フラグを指定できます。指定フラグには、以下のような意味があります。

**a** ファイル・アクセス時間を保存しない

**e**

ユーザー ID、グループ ID、ファイル・モード、アクセス時間、変更時間、および ACL を保存する

**m** ファイル変更時間を保存しない

**o** ユーザー ID とグループ ID を保存する

**p** ファイル・モードを保存する

**-e** フラグも **-o** フラグも指定しない場合、またはユーザー ID とグループ ID がいずれの理由でも保存されない場合、**pax** コマンドは、ファイル・モードの **S\_ISUID** ビットと **S\_ISGID** ビットを設定しません。これらの項目のいずれかの保存が失敗すると、**pax** コマンドは診断メッセージを標準エラーに書きます。項目のいずれかの保存に失敗すると、終了状況に影響を及ぼしますが、取り出されたファイルが削除されることはありません。指定されたフラグが重複するか互いに矛盾する場合は、最後に指定されたフラグが優先されます。例えば、**-p eme** が指定されると、ファイル変更時間が保存されます。

### **-r**

標準入力からアーカイブ・ファイルを読み取ります。

**-s** *ReplacementString*

**ed** コマンドの構文を使い、置換式 *ReplacementString* に従って、*Pattern* パラメーターまたは *File* パラメーターで指定したファイル・メンバーやアーカイブ・メンバーの名前を変更します。置換式は、以下のようなフォーマットになります。

**-s** */old/new[gp]*

ここで (**ed** コマンドの場合と同様に)、*old* は基本正規表現であり、*new* には **&** (アンパーサンド)、**\n** (**n** は数値) 逆参照、または副次式のマッチングを入れることができます。*old* 文字列には改行文字も入れることができます。

すべての非 **null** 文字を区切り文字として使用できます (例では **\** (バックスラッシュ) が区切り文字です)。複数の **-s** フラグ式を指定できます。この式は、指定した順序で適用され、最初に成功した置換で終わります。オプションのフッター **g** 文字は **ed** コマンドの場合と同様に機能します。オプションのフッター **p** 文字を使用すると、成功した置換が標準エラーに書き込まれます。空文字列と置換されるファイルまたはアーカイブ・メンバー名は、アーカイブの読み取りおよび書き込み時には無視されます。

**-t**

入力ファイルのアクセス時間を、**pax** コマンドによって読まれる以前と同じ時間にします。

**-U**

ACL および拡張属性のアーカイブと抽出を行います。属性にはアクセス制御リスト (ACL) も含まれます。ACL タイプがターゲット・ファイルシステムでサポートされない場合は、ターゲット・ファイルシステムでサポートされる ACL タイプに変換されます。EA (拡張属性) がファイルシステムでサポートされない場合は、EA はコピーされません。アーカイブのメンバーをリストする場合、このオプションは、名前付き拡張属性があればその名前と、各ファイルに関連する ACL でアーカイブ・イメージの一部であるものがあればそのタイプを、リストします。

**-u**

同じ名前を持つ既存のファイルまたはアーカイブ・メンバーより古いファイルは無視します。

- ファイルを取り出すときは、アーカイブ・メンバーがファイルより新しいものであれば、ファイルシステムにあるファイルと同じ名前のアーカイブ・メンバーが取り出されます。
- ファイルをアーカイブ・ファイルに書き込むときは、ファイルがアーカイブ・メンバーより新しいものであれば、ファイルシステムにあるファイルと同じ名前のアーカイブ・メンバーが取り替えられます。**-a** フラグを指定した場合、これは、アーカイブに付加されることによって行われます。このフラグが指定されていない場合、アーカイブ内で実際に取り替えられるか、アーカイブに付加されるかは、未定です。
- ファイルを出力先ディレクトリーにコピーするときは、ソース階層内のファイルの方が新しければ、出力先階層のファイルがソース階層のファイルによって、あるいはソース階層のファイルとのリンクによって、置き換えられます。

**-v**

プロセスに関する情報を書き込みます。**-r** フラグも **-w** フラグも指定しない場合は、**-v** フラグが詳細な目次を作成します。そうでなければ、アーカイブ・メンバー・パス名が標準エラーに書き込まれます。

**-w**

指定されたアーカイブ・フォーマットで標準出力へファイルを書き込みます。

## -x Format

**ustar** であるデフォルト・フォーマットを使って、出力アーカイブ・フォーマットを指定します。**pax** コマンドは、以下のフォーマットを認識します。

**pax** **pax** 交換フォーマット。文字特殊アーカイブ・ファイル用のこのフォーマットのデフォルト・ブロック化値は 10240 です。512 増分の 512 から 32256 までのブロック化値がサポートされます。

**cpio** 拡張 **cpio** 交換フォーマット。文字特殊アーカイブ・ファイル用のこのフォーマットのデフォルト・ブロック化値は 5120 です。512 増分の 512 から 32256 までのブロック化値がサポートされます。

**ustar** 拡張 **tar** ファイルを圧縮する交換フォーマット。文字特殊アーカイブ・ファイル用のこのフォーマットのデフォルト・ブロック化値は 10240 です。512 増分の 512 から 32256 までのブロック化値がサポートされます。

既存のアーカイブ・フォーマットと異なるフォーマットでアーカイブ・ファイルに追加しようとする、**pax** コマンドは 0 以外の終了状況ですぐに終了します。

コピー・モードでは、**-x** フォーマットが無指定の場合、**pax** は **-x pax** が指定された場合と同様の働きをします。

## -X

パス名で指定したファイル階層を移動するとき、**pax** コマンドは異なるデバイス ID を持つディレクトリーへ降りることはありません。

## フラグの相互作用および処理順序

ファイルまたはアーカイブ・メンバーの名前で動作するフラグ (**-c**、**-i**、**-n**、**-s**、**-u**、および **-v**) は、以下のように相互作用します。

- ファイルを取り出す場合、アーカイブ・メンバーは、**-c**、**-n**、**-u** フラグで変更されたユーザー指定の *pattern* パラメーターに従って選択されます。次に、**-s** および **-i** フラグは、その順序で、選択されたファイルの名前を変更します。**-v** フラグは、それらの変更の結果作成された名前を書き込みます。
- ファイルをアーカイブ・ファイルに書き込む場合、またはファイルをコピーする場合は、**-n** フラグ (コピー・モードの場合はこのオプションは無効) と **-u** フラグで変更されたユーザー指定のパス名に従って、ファイルが選択されます。次に、**-s** および **-i** フラグは、その順序で、それらの変更の結果作成された名前を変更します。**-v** フラグは、その変更の結果作成された名前を書き込みます。
- **-u** フラグと **-n** フラグの両方を指定すると、**pax** コマンドは、選択したファイルがそれを比較するファイルより新しくない限り、そのファイルを選択されたものと見なしません。

## リスト・モード・フォーマットの仕様

**-o listopt=format** オプションを指定したリスト・モードでは、フォーマット引数は、選択された各ファイルごとに適用されます。**pax** は、選択された各ファイルごとに **listopt** 出力に改行文字を追加します。フォーマット引数は、**printf()** で説明されたフォーマット文字列として使用されますが、次の例外がありません。

1. シーケンス *keyword* は、フォーマット変換指定子の前に置くことができます。変換引数は、*keyword* の値によって定義されます。サポートされるキーワードは次のとおりです。
  - **ustar** および **cpio** ヘッダー・ブロックの任意のフィールド名エントリー。
  - 拡張ヘッダーに定義された任意のキーワード、または拡張ヘッダー内の拡張として提供された任意のキーワード。

例えば、シーケンス `%(charset)s` は、拡張ヘッダー内の文字集合の名前の文字列値です。

キーワード変換引数の結果は、末尾 `NULL` のない、適用可能ヘッダー・フィールドまたは拡張ヘッダーからの値です。

変換引数として使用されるキーワードと値の組はすべて、UTF8 エンコードから、適用可能なローカル・ファイルシステム、ユーザー・データベースなどに適した文字集合に変換されます。

2. 追加の変換文字 **T** は、時刻形式を指定します。 **T** 変換文字の前にはシーケンス `keyword=subformat` を付けることができます。この場合、`subformat` は、**date** コマンドによって許可された日付形式です。デフォルト・キーワードは **mtime** で、デフォルト・サブフォーマットは `%b %e %H:%M %Y` です。
3. 追加の変換文字 **M** は、**ls -l** コマンドによって表示されるファイル・モード文字列を指定します。`keyword` を省略すると、**mode** キーワードが使用されます。例えば、`%1M` は、**ls -l** コマンドの `entry type` フィールドに対応する 1 文字を書き込みます。
4. 追加の変換文字 **D** は、ブロックまたはスペシャル・ファイルが適用可能であれば、そのデバイスを指定します。これが適用可能でない場合に `keyword` を指定すると、この変換は `%keyword u` と同じになります。適用可能でない場合に `keyword` を指定しないなら、この変換は `<スペース>` と同じになります。
5. 追加の変換文字 **F** は、パス名を指定します。 **F** 変換文字の前には、コンマで区切られたキーワードのシーケンスを付けることができます。

`keyword,keyword...`

`null` でない全キーワードの値は連結され、それぞれが `/` で区切られます。デフォルトは、キーワード・パスが定義されている場合は `path` です。それ以外の場合、デフォルトは `prefix,name` です。

6. 追加の変換文字 **L** は、シンボリック・リンク拡張を指定します。現在のファイルがシンボリック・リンクである場合、`%L` は次のように拡張されます。

`"%s -> %s", value_of_keyword, contents_of_link`

シンボリック・リンクでない場合、`%L` 変換文字は `%F` と同じです。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0** 正常終了。
- >0** エラーが発生しました。

## 例

1. `olddir` ディレクトリ階層を `newdir` にコピーするには、以下のように入力します。

```
mkdir newdir
pax -rw olddir newdir
```

2. 現行ディレクトリの内容をテープ・ドライブにコピーするには、以下のように入力します。

```
pax -wf /dev/rmt0
```

3. ファイル `xxx` を `XXX` としてアーカイブし、正常に置き換えられたものを表示するには、以下のように入力します。

```
pax -wvf/dev/rfd0 -s /xxx/XXX/p xxx
```

または

```
pax -wvf/dev/rfd0 -s/x/X/gp xxx
```

- 標準入力からファイルを読み、指定したサイズでデータ・ストリーム・ファイルにダンプするには、以下のように入力します。

```
dd if=/dev/hd6 bs=36b count=480 | pax -wf /dev/rfd0 -o  
datastream=_filename_,datastr_size=_size_
```

- アーカイブ **pax.ar** の中のファイルを指定した形式でリスト表示するには、次のように入力します。

```
pax -v -o listopt="start %F end" -f pax.ar
```

- pax** 形式のアーカイブ **pax.ar** を作成するには、次のように入力します。

```
pax -wf pax.ar -x pax file1
```

- pax** 形式のアーカイブ **pax.ar** から新しいパスにファイルを抽出するには、次のように入力します。

```
pax -rvf pax.ar -x pax -o path=newfilename
```

- シンボリック・リンクの内容を、ソースから宛先にコピーするには、次のように入力します。

```
pax -rwL srclink destdir
```

- アーカイブからグループ名 **bin** のファイルを抽出するには、次のように入力します。

```
pax -rvf pax.ar -x pax -o gname=bin
```

- pax** 形式のアーカイブからの抽出において、パス名を無視するには、次のように入力します。

```
pax -rvf pax.ar -o delete=path
```

- アーカイブの作成時に、長いユーザー名およびグループ名が切り捨てられないようにするには、次のように入力します。

```
pax -wEf file.pax file
```

- olddir** ディレクトリ階層を、ファイルに関連する **ACL** および **EA** 付きで **newdir** にコピーするには、次のように入力します。

```
mkdir newdir
```

```
pax -rUw olddir newdir
```

## ファイル

**/usr/bin/pax**                    **pax** コマンドが入っています。

## 関連情報

**ed** コマンド、**cpio** コマンド、**tar** コマンド、および **rmt** コマンド。

オペレーティング・システムおよびデバイスのマネージのファイルのセクション。

---

## pcat コマンド

### 目的

ファイルをアンパックして、標準出力に書き出します。

### 構文

**pcat** *File...*

## 説明

**pcat** コマンドは、*File* パラメーターで指定されたファイルを読み、アンパックし、それらを標準出力に書き出します。指定したファイルが **.z** 文字で終わっているかどうかに関係なく、**pcat** コマンドはそのファイルがパックされているものと見なしてアンパックします。

**pcat** コマンドの終了値は、アンパックできなかったファイルの数です。以下のいずれかが生じた場合には、ファイルをアンパックできません。

- ファイル名 (**.z** を除く) が 253 バイトを超える場合。
- ファイルをオープンできない場合。
- ファイルがパック・ファイルでない場合。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0 正常終了。
- >0 エラーが発生しました。

## 例

1. 圧縮ファイルを表示するには、以下のように入力します。

```
pcat chap1.z chap2 | pg
```

このコマンド・シーケンスは、圧縮ファイル `chap1.z` と `chap2.z` を一度に 1 ページずつ ( `| pg` ) 拡張フォーマットで画面に表示します。**pcat** コマンドは、`.z` が付いているかどうかに関係なくファイルを受け入れるので注意してください。

2. ディスクに格納したコピーを拡張しないで圧縮ファイルを使用するには、以下のように入力します。

```
pcat chap1.z | grep 'Greece'
```

このコマンド・シーケンスは、**pcat** コマンドによって `chap1.z` の内容が拡張フォーマットで表示されないようにして、**grep** コマンドにパイプ接続します。

## ファイル

`/usr/bin/pcat` **pcat** コマンドが入っています。

## 関連情報

**cat** コマンド、**grep** コマンド、**pack** コマンド、**unpack** コマンド。

オペレーティング・システムおよびデバイスのマネージのファイルおよび入出力リダイレクトのセクション。

---

## pdelay コマンド

### 目的

遅延ログイン・ポートを使用可能にするか、またはその使用可能性を報告します。



## 構文

**pdelay** [ -a ] [ *Device* ]

## 説明

**pdelay** コマンドは、遅延ポートを使用可能にします。遅延ポートは共用ポートと同様に使用可能になりますが、ただし 1 文字か複数の文字 (通常は復帰) を入力するまでは **login herald** (ログイン・ヘラルド) は表示されません。ポートがリモート・システムに直接接続されるか、インテリジェント・モデムに接続される場合、遅延ポートとして使用可能になり、**getty** コマンドがリモート側にある **getty** と、またはローカル接続のモデムと会話するのを防止します。このアクションはシステム・リソースを保存し、**pdelay enabled=delay** に等価なものです。 *Device* パラメーターを指定しない場合には、**pdelay** コマンドが現行で使用可能になっているポート名を報告します。

*Device* パラメーターを使用して、使用可能にするポートを指定します。以下の値が使えます。

- **F/dev/tty1** デバイスなどの完全デバイス名。
- **tty1** デバイスなど、単純デバイス名。
- 番号 (例えば、**/dev/tty1** デバイスを示す 1 など)。

注: このコマンドを実行するには root ユーザー権限を持っていないければなりません。

## フラグ

**-a** すべてのポートを遅延状態として使用可能にします。

## 例

現在使用可能な遅延ポートの名前を表示するには、以下のように入力します。

```
pdelay
```

## ファイル

**/etc/locks**

**pshare** および **pdelay** コマンドの **lock** ファイルが入っています。

**/usr/sbin/pdelay**

**pdelay** コマンドが入っています。

## 関連情報

**getty** コマンド、**init** コマンド、**pdisable** コマンド、**penable** コマンド、**phold** コマンド、**pshare** コマンド、**pstart** コマンド。

---

## pdisable コマンド

### 目的

ログイン・ポートを使用不可にします。

### 構文

**pdisable** [ -a ] [ *Device* ]



## 説明

**pdisable** コマンドは、ユーザーが特定のポートにログインしていても、そのポートを使用不可にします。システムは、**/etc/inittab** ファイルのエントリを更新し、次に **init** プロセスにシグナルを送信することによって、ポートを使用不可にします。**init** プロセスがシグナルを受信して、更新された状況エントリを読み込んだとき、適切なアクションを取ります。

*Device* パラメーターを使用して、使用不可にするポートを指定します。以下の値が使えます。

- **/dev/tty1** デバイスなどの完全デバイス名。
- **tty1** デバイスなどの単純デバイス名。
- 番号 (例えば、**/dev/tty1** デバイスを示す 1 など)。

*Device* パラメーターを指定しない場合、**pdisable** コマンドがその設定において現在使用不可になっているポート名を報告します。

注: このコマンドを実行するには root ユーザー権限を持っていないければなりません。

## フラグ

**-a** 現在使用可能になっているすべてのポートを使用不可にします。

## 例

1. 現在使用不可になっているすべてのポート名を表示するには、以下のように入力します。

```
pdisable
```

2. ユーザーがログインしていても、使用可能になっているすべてのポートを使用不可にするには、以下のように入力します。

```
pdisable -a
```

3. **/dev/tty8** ポートに接続したワークステーションを使用不可にするには、以下のように入力します。

```
pdisable tty8
```

## ファイル

**/etc/locks**

**/usr/sbin/pdisable**

**pshare** および **delay** コマンドの **lock** ファイルが入っています。

**pdisable** コマンドが入っています。

## 関連情報

**init** コマンド、**pdelay** コマンド、**penable** コマンド、**phold** コマンド、**pshare** コマンド、**pstart** コマンド。

**inittab** ファイル。

---

## penable コマンド

### 目的

ログイン・ポートを使用可能にするか、使用可能性を報告します。

## 構文

**penable** [ **-a** ] [ *Device* ]

## 説明

**penable** コマンドは、通常のポートを使用可能にします。通常ポートは非同期なので、ユーザーはログインできるだけです。ポートが使用可能になっている間は、ポートの発信使用はできません。システムでは、**/etc/inittab** ファイル内のエントリーを更新し、次に **init** プロセスにシグナルを送信することによって、ポートを使用可能にします。シグナルを受信して更新された状況エントリーを読み込むと、プロセスは、適切なアクションを取ります。

*Device* パラメーターを使用して、使用可能にするポートを指定します。以下の値が使えます。

- **F/dev/tty1** デバイスなどの完全デバイス名。
- **tty1** デバイスなど、単純デバイス名。
- 番号 (例えば、**/dev/tty1** デバイスを示す 1 など)。

*Device* パラメーターを指定しない場合、**penable** コマンドが現在使用可能な通常ポート名を報告します。

注: このコマンドを実行するには root ユーザー権限を持っていないければなりません。

## フラグ

**-a** すべての通常ポートを使用可能にします。

## 例

**/etc/inittab** ファイル内にリストされたすべての通常ポートを使用可能にするには、以下のように入力します。

```
penable -a
```

## ファイル

**/etc/locks** **pshare** および **pdelay** コマンドの **lock** ファイルが入っています。  
**/usr/sbin/penable** **penable** コマンドが入っています。

## 関連情報

**init** コマンド、**pdelay** コマンド、**pdisable** コマンド、**phold** コマンド、**pshare** コマンド、**pstart** コマンド。

**inittab** ファイル。

---

## perfwb コマンド

### 目的

Performance Workbench を開始して、システム稼働状況をモニターします。

## 構文

### perfwb

注: DISPLAY 環境変数が設定されている必要があります。

## 説明

**perfwb** コマンドは、Performance Workbench を開始するために使用されます。Performance Workbench は、システム稼働状況およびプロセスをモニターするためのグラフィカル・インターフェースです。

パネルは、区画構成および CPU とメモリーの使用量を表示します。

別のパネルでは上位のプロセスがリストされます。プロセスは、さまざまに提供されるメトリックに基づいてソートすることが可能です。特定のプロセスのみにリストを制限する、フィルター・デバイスも提供されます。

## 終了状況

0            コマンドは正常に実行されました。  
>0          エラーが発生しました。

## 位置

/usr/bin/perfwb

## ファイル

/usr/bin/perfwb  
\$HOME/workspace

**perfwb** コマンドが入っています。  
設定が入っている **perfwb** 作業ディレクトリーが入っています。

## 関連情報

561 ページの『ps コマンド』、**topas** コマンド。

---

## pg コマンド

### 目的

ファイルをディスプレイに合わせてフォーマットします。

### 構文

**pg** [ - *Number* ] [ -c ] [ -e ] [ -f ] [ -n ] [ -p *String* ] [ -r ] [ -s ] [ +*LineNumber* ] [ +*PatternI* ] [ *File* ... ]

### 説明

**pg** コマンドは *File* パラメーターからファイル名を読み取り、そのファイルを一度に 1 画面ずつ標準出力に書き出します。- (ダッシュ) を *File* パラメーターとして指定する場合、またはオプションを付けずに **pg** コマンドを実行する場合、**pg** コマンドを使用して標準入力を読み込みます。各画面には、プロンプト

が表示されます。Enter キーを押した場合、別のページが表示されます。 **pg** コマンドと一緒にサブコマンドを使うと、ファイル内のデータの見直し、または検索を行うことができます。

ワークステーションの属性を決めるために、**pg** コマンドを使用して、**TERM** 環境変数で指定されたワークステーション・タイプのファイルをスキャンします。デフォルトは、**dumb** タイプです。

**pg** コマンドが一時停止して、プロンプトを出したら、サブコマンドを入力できます。これらのサブコマンドの中には、表示画面をファイルの特定のロケーションに変更するもの、テキスト内で特定パターンの検索のいくつかを変更するもの、**pg** コマンドが機能する環境を変更するものがあります。

## ファイル内の位置の変更

以下のサブコマンドは、ファイル内の選択した個所を表示します。

<i>Page</i>	<i>Page</i> パラメーターで指定したページを表示します。
<b>+Number</b>	<i>Number</i> の値を加えたページを現在のページに表示します。
<b>-Number</b>	現在のページから <i>Number</i> の値を引いたページを表示します。
<b>l</b>	(L の小文字) ディスプレイを下方に 1 行スクロールします。
<i>Numberl</i>	<i>Number</i> パラメーターで指定した行を画面の最上部に表示します。
<b>+Numberl</b>	指定された行数だけ画面表示を下方にスクロールします。
<b>-Numberl</b>	指定された行数だけ画面表示を上方にスクロールします。
<b>d</b>	画面の半分を下方にスクロールします。Ctrl-D キー・シーケンスを押すと、 <b>d</b> サブコマンドと同様に機能します。
<b>-d</b>	画面の半分を上方にスクロールします。-Ctrl-D キー・シーケンスを押すと、 <b>-d</b> サブコマンドと同様に機能します。
<b>Ctrl-L</b>	現在のページを再表示します。1 つの . (ドット) は、 <b>Ctrl-L</b> キー・シーケンス・サブコマンドと同様に機能します。
<b>\$</b>	ファイル内の最後のページを表示します。パイプラインからの入力には、このサブコマンドは使用しないでください。

## テキスト・パターンの検索

以下のサブコマンドは、テキスト内のテキスト・パターンを検索します。( **ed** コマンドに記述されたパターンも使用できます。) サブコマンドは、**-n** フラグを使用した場合でも、必ず改行文字で終わらなければなりません。

[k.a-z]k. などの式では、マイナスは、現在の照合順序に従って a から z などのある範囲を暗黙指定します。照合順序は、文字範囲での使用のための等価クラスを定義します。

**[Number]/Pattern/** *Number* 変数で指定した *Pattern* の値のオカレンスを検索します。検索は、現行ページの直後から始まり、現行ファイルの終わりまで循環しないで続きます。 *Number* 変数のデフォルトは 1 です。

**Number?Pattern?**

**Number^Pattern^**

*Number* 変数で指定した *Pattern* の値のオカレンスを逆方向に向かって検索します。検索は、現行ページの直前から始まり、現行ファイルの冒頭まで循環なしに続きます。 *Number* 変数のデフォルトは 1 です。 ^ 表記は、? 表記を適切に処理しない端末装置を 100 台追加するのに便利です。

検索後、**pg** コマンドを指定すると、画面の最上部に一致するパターンを持つ行が表示されます。検索コマンドに接尾部 **m** または **b** を追加することによって、表示位置を変更できます。接尾部 **m** を指定する

と、後続のどのサブコマンドの実行時にも画面の中央に、一致するパターンを持つ行が表示されます。接尾部 **b** を指定すると、後続のどのサブコマンドの実行時にも画面の下部に一致するパターンを持つ行が表示されます。接尾部 **t** を指定すると、画面の最上部に一致するパターンを持つ行が再表示されます。

## pg 環境の変更

以下のサブコマンドを使用して、**pg** コマンドの環境を変更できます。

<b>[Number]n</b>	<i>Number</i> 変数で指定した数だけ、コマンド行の次のファイルの検査を開始します。 <i>Number</i> 変数のデフォルトは、1 番目です。
<b>[Number]p</b>	<i>Number</i> 変数で指定した数だけ、コマンド行の直前のファイルの検査を開始します。 <i>Number</i> 変数のデフォルトは、1 番目です。
<b>[Number]w</b>	テキストの別のウィンドウを表示します。 <i>Number</i> 変数を指定すると、ウィンドウ・サイズが指定した行数に設定されます。このサブコマンドは、 <b>[Number]z</b> サブコマンドと同じです。
<b>[Number]z</b>	テキストの別のウィンドウを表示します。 <i>Number</i> 変数を指定すると、ウィンドウ・サイズが指定した行数に設定されます。このサブコマンドは、 <b>[Number]w</b> サブコマンドと同じです。
<b>s File</b>	指定したファイルに入力データを保存します。検査しようとする現行ファイルだけが保存されます。 <b>-n</b> フラグを指定した場合でも、このコマンドは、必ず改行文字で終わらなければなりません。
<b>h</b>	使用可能なサブコマンドを省略した要約を表示します。
<b>q</b> または <b>Q</b>	<b>pg</b> コマンドを終了します。
<b>!</b> コマンド	指定したコマンドを <b>SHELL</b> 環境変数で指定したシェルに送ります。これを使用できない場合には、デフォルトのシェルが使われます。 <b>-n</b> フラグを使用した場合でも、このコマンドは、必ず改行文字で終わらなければなりません。

### 重要:

- キー・シーケンス **QUIT WITH DUMP** (Ctrl-\) または **INTERRUPT** (Ctrl-C) を押すと、出力の一部が失われます。これは、**QUIT** シグナルを受信したときに、出力キューで待機している文字が削除されるためです。
- ワークステーションのタブを 8 桁ごとに設定しないと、予想外の結果になる恐れがあります。

ワークステーションに出力を送信しているときはいつでも、キー・シーケンス **QUIT WITH DUMP** または **INTERRUPT** を押すことができます。こうすると、**pg** コマンドを指定することによって、出力の送信が停止し、プロンプトが表示されます。その場合は、コマンド・プロンプトから上記のサブコマンドのいずれかを入力できます。

標準出力がワークステーションでない場合、それぞれのファイルの前にヘッダーが表示されることを除いて、**pg** コマンドは、**cat** コマンドと同じ動作をします。

ワークステーション入力の待機中に、キー・シーケンス **INTERRUPT** を押すと、**pg** コマンドは実行を停止します。プロンプト間でこれらのシグナルは現行タスクに割り込み、ユーザーはプロンプト・モードに置かれます。

## フラグ

<b>-c</b>	カーソルをホーム・ポジションに移動し、それぞれのページの前の画面をクリアします。このフラグが無視されるのは、 <b>terminfo</b> ファイルのワークステーション・タイプに、 <code>clear_screen</code> フィールドが定義されていない場合です。
<b>-e</b>	各ファイルの終わりで一時停止しません。
<b>-f</b>	行を分割しません。通常、 <b>pg</b> コマンドは画面幅よりも長い行を分割します。

<b>-n</b>	<b>pg</b> コマンド文字を入力すると処理を停止します。通常、コマンドは改行文字で終わらせなければなりません。
<b>-p String</b>	指定した文字列をプロンプトとして使用します。 <i>String</i> に <b>%d</b> 値が含まれる場合、その値はプロンプトの現行ページ番号で置き換えられます。デフォルトのプロンプトは、 <b>:</b> (コロン) です。指定した文字列にスペースが含まれる場合、文字列を引用符で囲まなければなりません。
<b>-r</b>	"!" サブコマンドが使用された場合に、シェル・エスケープを防ぎます。
<b>-s</b>	すべてのメッセージとプロンプトを強調表示します。
<b>+LineNumber</b>	指定した行番号から始めます。
<b>-Number</b>	ウィンドウ内の行数を指定します。 24 行入るワークステーションの場合、デフォルトは 23 です。
<b>+IPatternI</b>	指定したパターンを含む最初の行から始めます。

## 終了状況

このコマンドは、以下の終了値を戻します。

<b>0</b>	正常終了。
<b>&gt;0</b>	エラーが発生しました。

## 例

ファイルの内容を 1 ページずつ見るには、以下のように入力します。

```
pg filename
```

## ファイル

<b>/usr/bin/pg</b>	<b>pg</b> コマンドが入っています。
<b>/usr/share/lib/terminfo/*</b>	端末タイプを定義する <b>terminfo</b> ファイルが入っています。
<b>/tmp/pg*</b>	<b>pg</b> コマンドを使用したときに作成される一時ファイルが入っています。

## 関連情報

**cat** コマンド、**ed** コマンド、**grep** コマンド。

オペレーティング・システムおよびデバイスのマネージのファイルとディレクトリーのアクセス・モードのセクション。

オペレーティング・システムおよびデバイスのマネージの入出力ダイレクトおよびシェルのセクション。

オペレーティング・システムおよびデバイスのマネージのファイルのセクション。

---

## phold コマンド

### 目的

保持中のログイン・ポートを使用不可にするか、その使用可能性を報告します。

### 構文

```
phold [ -a ] [ Device ]
```

## 説明

**phold** コマンドは 1 組のログイン・ポートを使用不可にします。 **phold** コマンドを使用すると、既にログインしているユーザーは継続できますが、そのほかのユーザーはログインできません。ユーザーは使用不可にされたポートにはログインできません。システムは、**/etc/inittab** ファイルのエントリを更新し、次に **init** プロセスにシグナルを送信することによって、ポートを使用不可にします。 **init** プロセスがシグナルを受信して、更新された状況エントリを読み込んだとき、適切なアクションを取ります。

*Device* パラメーターを使用して、使用不可にするポートを指定します。以下の値が使えます。

- **/dev/tty1** デバイスなどの完全デバイス名。
- **tty1** デバイスなどの単純デバイス名。
- 番号 (例えば、**/dev/tty1** デバイスを示す 1 など)。

*Device* パラメーターを指定しない場合は、**phold** コマンドがその設定において現在使用不可になっているポート名を報告します。

注: このコマンドを実行するには **root** ユーザー権限を持っていないければなりません。

## フラグ

**-a** 現在使用可能になっているすべてのポートを保持します。

## 例

現在保持しているポートをリストするには、以下のように入力します。

```
phold
```

## ファイル

**/etc/locks** **pshare** および **pdelay** コマンドの **lock** ファイルが入っています。  
**/etc/phold** **phold** コマンドが入っています。

## 関連情報

**init** コマンド、**pdelay** コマンド、**pdisable** コマンド、**penable** コマンド、**pshare** コマンド、**pstart** コマンド。

**inittab** ファイル。

---

## pic コマンド

### 目的

描画するために **troff** コマンドの入力のプリプロセスをします。

### 構文

```
pic [ -T Name ] [ -l File... ]
```



## 説明

**pic** コマンドは、組版システムで簡単な図を描くための **troff** コマンド・プリプロセッサです。基本的なオブジェクトは、ボックス、円、楕円、直線、スプライン、矢印、弧、および *Text* 変数で指定したテキストです。最高レベルのオブジェクトが、ピクチャーです。

*File* 描画するために **pic** コマンドが処理する **troff** コマンドからの出力を指定します。

## ピクチャー

**pic** コマンド内の最上位のオブジェクトがピクチャーです。

**.PS** *OptionalWidth OptionalHeight*

*ElementList*

**.PE**

**.PF** マクロを **.PE** の代わりに使用した場合、表示後の位置が入力時の状態に復元されます。

*OptionalWidth* ピクチャーがあれば、内部で使用される寸法に関係なく、その幅をインチ単位で指定します。最大値は 8.5 です。

*OptionalHeight* 同じ比率にスケールされる高さの値をインチ単位で指定します。この値は、デフォルトとは異なります。最大値は 14 です。

*ElementList* 以下のようなエレメントのリストを表示します。

*Shape AttributeList*

*For Statement*

*Placename: Element*

*If Statement*

*Placename: Position*

*Copy Statement*

*Variable = Expression*

*Print Statement*

*Direction*

*Plot Statement*

*{ List of Elements }*

*sh X Commandline X*

*[ List of Elements ]*

*troff-command*

変数名は、小文字で始まり、ゼロか複数の文字または数字が続きます。位置名は大文字で始まり、ゼロまたは複数の文字か数字が続きます。位置名と変数名は、1 つのピクチャーから次のピクチャーへと、その値を保存します。

リスト内のエレメントは、改行文字か、; (セミコロン) で区切る必要があります。長いエレメントを続けるには、行を \ (バックスラッシュ) で終了します。コメントは、# 文字で始め、改行文字で終了します。

## 図形エレメント

図形エレメント・オブジェクトは、以下のとおりです。

**box**

circle  
 ellipse  
 arc  
 line  
 arrow  
 spline  
 move

*Text-List*

**arrow** オブジェクトは、 **->** 属性を持つ **line** オブジェクトと同じです。

**属性** *AttributeList* エレメントは、ゼロ個またはそれ以上の属性を順に並べたものです。各属性は、値が続く場合もあるキーワードで構成されます。

属性	属性
<b>h(eigh)t</b> <i>Expression</i>	<b>wid(th)</b> <i>Expression</i>
<b>rad(ius)</b> <i>Expression</i>	<b>diam(eter)</b> <i>Expression</i>
<b>up</b> <i>OptionalExpression</i>	<b>down</b> <i>OptionalExpression</i>
<b>right</b> <i>OptionalExpression</i>	<b>left</b> <i>OptionalExpression</i>
<b>from</b> <i>Position</i>	<b>to</b> <i>Position</i>
<b>at</b> <i>Position</i>	<b>with</b> <i>Corner</i>
<b>by</b> <i>Expression, Expression</i>	<b>then</b>
<b>dotted</b> <i>OptionalExpression</i>	<b>dashed</b> <i>OptionalExpression</i>
<b>chop</b> <i>OptionalExpression</i>	<b>-&gt; &lt;- &lt;-&gt;</b>
<b>invis</b>	<b>same</b>
<i>Text-list</i>	

存在しない属性と値は、デフォルトから充てんされます。すべての属性がすべての図形エレメントについて意味があるわけではありません。無関係な属性は処理されません。現在、意味のある属性は以下のとおりです。

図形エレメント	属性
<b>box</b>	<b>h(eigh)t, wid(th), at, same, dotted, dashed, invis, Text</b>
<b>circle, ellipse</b>	<b>rad(ius), diam(eter), h(eigh)t, wid(th), at, same, invis, Text</b>
<b>arc</b>	<b>up, down, left, right, h(eigh)t, wid(th), from, to, at, rad(ius), invis, ccw, cw, &lt;-, -&gt;, &lt;-&gt;, Text</b>
<b>line, arrow</b>	<b>up, down, left, right, h(eigh)t, wid(th), from, to, by, then, at, same, dotted, dashed, invis, &lt;-, -&gt;, &lt;-&gt;, Text</b>
<b>spline</b>	<b>up, down, left, right, h(eigh)t, wid(th), from, to, by, then, at, same, invis, &lt;-, -&gt;, &lt;-&gt;, Text</b>
<b>move</b>	<b>up, down, left, right, to, by, same, Text</b>
<i>Text-list</i>	<b>at, Text-item</b>

**at** 属性は、指定位置に図形の中心を置くことを意味します。行、スプライン、および弧の場合、**h(eigh)t** および **wid(th)** 属性が矢印の先頭のサイズを示します。

*Text-item* 変数は、通常、何らかの図形エレメントの属性です。デフォルトでは、その変数はオブジェクトの図形の中心に置かれます。スタンドアロン・テキストのみでも受け入れられます。 *Text-list* の図形エレメントは、テキスト・エントリーのリストです。テキスト・エントリーは、以下のようにオプションとして位置決め要求が続く引用符で囲まれた文字列です。

"..."

"..." **center**

"..." **ljust**

"..." **rjust**

"..." **above**

"..." **below**

一部の図形エレメントに複数のテキスト・エントリーがあると、修飾されている場合を除いて、垂直に中央揃えされます。位置決め要求は各エントリーに独自に適用されます。

テキスト・エントリーには、サイズおよびフォントの変更やローカルの動作などを管理する **troff** コマンドが含まれています。これらのコマンドのバランスをとって、入力状態が終了前の状態に復元されることを確認します。

## 位置/ロケーション

1 つの位置は最終的には X,Y の座標対ですが、以下のような方法で指定することもできます。

*Place*

(*Position* )

*Expression*, *Expression*

(*Position* ) [**+/-** (*Expression*, *Expression*)]

(*Position* ) [**+/-** *Expression*, *Expression*]

(*Place1*, *Place2* )

(*Place1.X*, *Place2.Y*)

*Expression* < *Position*, *Position* >

*Expression* [**of the way**] **between** *Position* **and** *Position*

*Placename* [*Corner*]

*Corner* *Placename*

### Here

*Corner* **of** *Nth* *Shape*

*Nth shape* [*Corner*]

注: *Corner* 変数は、以下のように 8 つのコンパス・ポイントのうちのいずれか、または図形エレメントの中心、始まり、終わりを指定します。

**.n .e .w .s .ne .se .nw .sw**

**.t .b .r .l**

**c .start .end**

ピクチャーの各オブジェクトは、序数を持ちます。*Nth* は、以下のように参照します。

- *Nth*
- *Nth last*

**pic** コマンドは、**1th** や **3th** などのような名前を受け入れられるほど柔軟なものです。**1st** や **3st** も同様に使用されます。

## 変数

組み込み変数およびそのデフォルト値は、以下のとおりです。

<b>boxwid</b>	0.75
<b>boxht</b>	0.5
<b>circlerad</b>	0.25
<b>arcrad</b>	0.25
<b>ellipsewid</b>	0.75
<b>ellipseht</b>	0.5

<b>linewidth</b>	0.5
<b>lineht</b>	0.5
<b>movewid</b>	0.5
<b>moveht</b>	0.5
<b>arrowwid</b>	0.05
<b>arrowht</b>	0.1
<b>textwid</b>	0
<b>textht</b>	0
<b>dashwid</b>	0.5
<b>scale</b>	1

これらのデフォルト値は、いつでも変更できます。また、新しい値は、もう一度変更されるまで、ピクチャー一間でそのまま有効です。

変数 **textht** および **textwid** は任意の値を設定して、位置決めを管理できます。生成されたピクチャーの幅と高さは、**.PS** マクロ行か個別に設定できます。[ (左大括弧) 区切り文字と ] (右大括弧) 区切り文字の間の変数を変更すると、ブロックを終了するとき前の値に戻ります。寸法は、出力中に **scale** で除算されます。

注: **.ps** (サイズ) 行によって、8 インチより大きいサイズが指定されたときでも、**pic** コマンドでは、生成され、**troff** に送られるピクチャーのサイズが 8x8 インチと制限されています。

## 式

以下の **pic** コマンドの式は、浮動小数点で評価されます。寸法を表す数字はすべて、インチ単位になります。

*Expression* + *Expression*

*Expression* - *Expression*

*Expression* \* *Expression*

*Expression* / *Expression*

*Expression* % *Expression* (modulus)

- *Expression*

(*Expression* )

**variable**

**number**

*Place* **.x**

*Place* **.y**

*Place* **.ht**

*Place* **.wid**

*Place* **.rad**

**sin(Expression) cos(Expression) atan2(Expression, Expression) log(Expression) sqrt(Expression)**  
**int(Expression) max(Expression, Expression) min(Expression, Expression) rand(Expression)**

## 論理演算子

**pic** コマンドは、論理計算用に以下の演算子を提供します。

<b>!</b>	否定 (ではない)
<b>&gt;</b>	よりも大きい
<b>&lt;</b>	よりも小さい
<b>&gt;/=</b>	よりも大きいか等しい
<b>&lt;/=</b>	よりも小さいか等しい
<b>&amp;&amp;</b>	And
<b> </b>	Or
<b>==</b>	等しい
<b>!=</b>	等しくない

## 定義

以下の **define** ステートメントは、文法の一部ではありません。

**define Name X Replacement text X**

*Replacement text* 変数での **\$1** や **\$2** のような値のオカレンスは、*Name* 変数が呼び出された場合、次のように対応するオプションで置き換えられます。

*Name*(*Option1*, *Option2*, ...)

存在しないオプションは、ヌル・ストリングで置き換えられます。 *Replacement text* 変数には、改行文字が使えます。

## copy および copy thru ステートメント

**copy** ステートメントには、以下のように直後に続くファイルまたは値からのデータが含まれています。

**copy** File

**copy thru** Macro

**copy** File **thru** Macro

**copy** File **thru** Macro **until** String

*Macro* パラメーター値には、定義済みマクロ名か、内容の一部でない何らかの文字で囲まれたマクロの内容が使えます。ファイル名を指定しないと、**copy** ステートメントは、次の **.PE** マクロ行までの入力をコピーします。

## for ループと if ステートメント

**for** ステートメントと **if** ステートメントは、以下のようにループと意思決定を規定します。

Variable=Expression **to** Expression **by** Expression **do** X anything X

**if** Expression **then** X anything X **else** X anything X

**by** と **else** 節は、オプションです。 **if** ステートメントの *Expression* 変数は、通常の関係演算子または *String1* **==** (または **!=**) *String2* 文字列検査を使用できます。

## その他の情報

**sh** コマンドは以下のようにコマンド行を実行します。

**sh** X Commandline X

式の値を以下のようにプロットできます。

**plot** Expression OptionalFormat Attributes

*Expression* 変数値は、評価されて文字列に変換されます (提供されている場合、フォーマット指定を使用します)。

充てんモードまたは非充てんモードの状態は、ピクチャーについて保存されます。

入力番号は **E** (指数) 表記で表すことができます。

## フラグ

**-TName** 指定したプリンター用の出力を作成します。Name には、以下の値を使用できます。

**ibm3812**

3812 ページ・プリンター。

**ibm3816**

3816 ページ・プリンター。

**hplj**

Hewlett-Packard LaserJet II。

**ibm5587G**

5587-G01 漢字プリンター (マルチバイト言語サポート)。

**psc**

PostScript® プリンター。

**X100**

AIXwindows ディスプレイ。

**X100K**

AIXwindows ディスプレイ (マルチバイト文字サポート)。

デフォルトは **ibm3816** です。

**注:** **troff** コマンドの **-TName** フラグを使用する代わりに、**TYPESETTER** 環境変数を上記のいずれかの値に設定してもかまいません。

- 標準入力に逆変換します。

## 関連情報

**grap** コマンド、**sh** コマンド、**troff** コマンド。

**.PE** マクロ、**.PF** マクロ、**.PS** マクロ、**me** マクロ・パッケージ。

---

## pick コマンド

### 目的

メッセージを内容によって選択し、その順序の作成および変更を行います。



## 構文

```
pick [ +Folder ] [ Messages ] [ -datefield Field ] [ -not ] [ -lbrace ] [ -after Date ] [ -before Date ]  
[ -cc "Pattern" ] [ -date "Pattern" ] [ -from "Pattern" ] [ -search "Pattern" ] [ -to "Pattern" ]  
[ -Component "Pattern" ] [ -rbrace ] [ -and ] [ -or ] [ -sequence Name [ -zero | -nozero ]  
[ -public | -npublic ] [ -list | -nolist ]
```

## 説明

**pick** コマンドは、特定の文字パターンまたは特定の日付のメッセージを選択します。 **-and**、**-or**、**-not**、**-lbrace**、および **-rbrace** フラグを使用して、メッセージ選択の複合条件を組み立てます。

## フラグ

<b>-after</b> <i>Date</i>	<i>Date</i> 変数によって指定された日付より後の日付のついたメッセージを選択します。 <i>Date</i> 変数には、以下の指定を使用します。 <table><tr><td><b>yesterday</b></td><td><b>today</b></td><td><b>tomorrow</b></td></tr><tr><td><b>sunday</b></td><td><b>monday</b></td><td><b>tuesday</b></td></tr><tr><td><b>wednesday</b></td><td><b>thursday</b></td><td><b>friday</b></td></tr><tr><td><b>saturday</b></td><td><b>-Days</b></td><td><b>SystemDate</b></td></tr></table>	<b>yesterday</b>	<b>today</b>	<b>tomorrow</b>	<b>sunday</b>	<b>monday</b>	<b>tuesday</b>	<b>wednesday</b>	<b>thursday</b>	<b>friday</b>	<b>saturday</b>	<b>-Days</b>	<b>SystemDate</b>
<b>yesterday</b>	<b>today</b>	<b>tomorrow</b>											
<b>sunday</b>	<b>monday</b>	<b>tuesday</b>											
<b>wednesday</b>	<b>thursday</b>	<b>friday</b>											
<b>saturday</b>	<b>-Days</b>	<b>SystemDate</b>											
<b>-and</b>	2 つのメッセージ選択フラグの間の論理 AND を形成します。例えば、 <b>pick -after Sunday -and -from mark</b> のようにします。 <b>-and</b> フラグは、 <b>-or</b> フラグより優先順位が高く、 <b>-not</b> フラグよりも優先順位が低くなります。 <b>-lbrace</b> および <b>-rbrace</b> フラグを使用して、この優先順位の指定を上書きできます。												
<b>-before</b> <i>Date</i>	指定した日付より前の日付を持つメッセージを選択します。 <i>Date</i> の指定方法は、 <b>-after</b> フラグを参照してください。												
<b>-cc</b> " <i>Pattern</i> "	cc: フィールドの " <i>Pattern</i> " 変数で指定した文字列を含むメッセージを選択します。												
<b>-date</b> " <i>Pattern</i> "	Date: フィールドの " <i>Pattern</i> " 変数で指定した文字列を含むメッセージを選択します。												
<b>-datefield</b> <i>Field</i>	<b>-after</b> および <b>-before</b> フラグを指定したときに、構文解析される日付フィールドを指定します。デフォルトでは、 <b>pick</b> コマンドは Date: フィールドを使用します。												
<b>+Folder</b>	選択したいメッセージを含むフォルダーを識別します。デフォルトでは、システムは現行フォルダーを使用します。												
<b>-from</b> " <i>Pattern</i> "	From: フィールドの " <i>Pattern</i> " 変数で指定した文字列を含むメッセージを選択します。												
<b>-help</b>	コマンド構文、使用可能なスイッチ (トグル)、およびバージョン情報をリストします。												
<b>-lbrace</b>	注: MH の場合は、このフラグ名を省略することはできません。 <b>-and</b> 、 <b>-or</b> 、および <b>-not</b> 演算をグループ化します。 <b>-lbrace</b> と <b>-rbrace</b> フラグの間の演算は、1 つの演算として実行されます。 <b>-lbrace</b> および <b>-rbrace</b> フラグをネストすることができます。												

<b>-list</b>	<p>選択したメッセージ番号のリストを標準出力に送信します。これにより、<b>pick</b> コマンドを使用して、メッセージ番号を生成して別のコマンドに対する入力として使用できます。例えば、火曜日より後に送信された現行フォルダーのメッセージをすべてスキャンするには、以下のように入力します。</p> <pre>scan 'pick -after tuesday -list'</pre>
<i>Messages</i>	<p>シーケンスを指定しない場合、デフォルトは、<b>-list</b> フラグです。</p> <p>検索するメッセージを指定します。複数のメッセージ、メッセージの範囲、または単一メッセージを指定できます。次のものを使用してメッセージを指定します。</p> <p><i>Number</i> メッセージの数。</p> <p><i>Sequence</i></p> <p>ユーザーが指定するメッセージのグループ。指定できる値は、以下のとおりです。</p> <ul style="list-style-type: none"> <li><b>all</b> フォルダー内のすべてのメッセージ。これはデフォルトです。</li> <li><b>cur</b> または <b>.</b> (ピリオド) 現行メッセージ。</li> <li><b>first</b> フォルダー内の最初のメッセージ。</li> <li><b>last</b> フォルダー内の最後のメッセージ。</li> <li><b>new</b> 新たに作成したメッセージ。</li> <li><b>next</b> 現行メッセージの次のメッセージ。</li> <li><b>prev</b> 現行メッセージの前のメッセージ。</li> </ul>
<b>-nolist</b>	<p><b>pick</b> コマンドが選択したメッセージ番号のリストを生成しないようにします。シーケンスが指定されていない場合、デフォルトは、<b>-nolist</b> フラグです。</p>
<b>-nopublic</b>	<p>シーケンスの使用に制限を与えます。<b>-nopublic</b> フラグは、シーケンスのメッセージではなく、シーケンスそのものを制限します。フォルダーが別のユーザーからの書き込みを保護されている場合、このオプションが、デフォルトです。</p>
<b>-not</b>	<p>2 つのメッセージ選択フラグの論理 NOT 演算を形成します。例えば、<b>pick -not -from george</b> などです。この構造体は、メッセージ選択フラグによって選択されたのではないメッセージすべてを実行します。<b>-not</b> フラグは、<b>-and</b> フラグより優先順位が高く、<b>-and</b> フラグは、<b>-or</b> フラグより優先順位が高くなります。</p> <p><b>-lbrace</b> および <b>-rbrace</b> フラグを使用して、この優先順位の指定を上書きできます。</p>
<b>-nozero</b>	<p>選択したメッセージを指定したシーケンスに追加します。</p>
<b>-or</b>	<p>2 つのメッセージ選択フラグの論理 OR 演算を形成します。例えば、<b>pick -from amy -or -from mark</b> などです。<b>-not</b> フラグは、<b>-and</b> フラグより優先順位が高く、<b>-and</b> フラグは、<b>-or</b> フラグより優先順位が高くなります。<b>-lbrace</b> および <b>-rbrace</b> フラグを使用して、この優先順位の指定を上書きできます。</p>
<b>-public</b>	<p>別のユーザーがシーケンスにアクセスできるようにします。<b>-public</b> フラグは、保護されたメッセージではなく、シーケンスそのものを使用できるようにします。フォルダーが別のユーザーによる書き込みから保護されていない場合、このオプションがデフォルトです。</p>
<b>-rbrace</b>	<p><b>-and</b>、<b>-or</b>、および <b>-not</b> 演算をグループ化します。<b>-lbrace</b> と <b>-rbrace</b> フラグの間の演算は、1 つの演算として実行されます。<b>-lbrace</b> および <b>-rbrace</b> フラグをネストすることができます。</p>
<b>-search "Pattern"</b>	<p>メッセージの任意の位置にある <i>"Pattern"</i> 変数で指定した文字列を含むメッセージを選択します。</p>

<b>-sequence</b> <i>Name</i>	<i>Name</i> 変数で指定したシーケンスで <b>pick</b> コマンドを指定して、選択したメッセージを格納します。
<b>-to</b> " <i>Pattern</i> "	To: フィールドの " <i>Pattern</i> " 変数で指定した文字列を含むメッセージを選択します。
<b>-zero</b>	指定したシーケンスをクリアしてから、選択したメッセージをシーケンスに入れます。このフラグがデフォルトです。
<b>-Component</b> " <i>Pattern</i> "	<i>Component</i> 変数で指定したヘッダー・フィールドの " <i>Pattern</i> " 変数で指定した文字列を含むメッセージを選択します。例えば、pick -reply-to amy などです。

## プロフィール・エントリー

次のプロフィール・エントリーは、*UserMHDDirectory/.mh\_profile* ファイルの一部です。

Current-Folder: デフォルトの現行フォルダーを設定します。  
Path: ユーザーの MH ディレクトリーを指定します。

## 例

1. ユーザーの jones から現行フォルダーのメッセージ番号リストを取得するには、以下のように入力します。

```
pick -from jones
```

システムは次のようなメッセージで応答します。

```
12
15
19
```

2. この 30 日以内に受信した schedule フォルダーのメッセージ番号リストを表示するには、以下のように入力します。

```
pick +schedule -after -30
```

システムは次のようなメッセージで応答します。

```
5
8
21
30
```

## ファイル

**\$HOME/.mh\_profile** ユーザーの MH プロファイルが入っています。  
**/usr/bin/pick** **pick** コマンドが入っています。

## 関連情報

**mark** コマンド。

**.mh\_alias** ファイル・フォーマット、**.mh\_profile** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージのメール・アプリケーションのセクション。

---

## ping コマンド

### 目的

ネットワーク・ホストにエコー要求を送信します。

### 構文

```
ping [ -d ] [ -D ] [ -n ] [ -q ] [ -r ] [ -v ] [ -R ] [ -a addr_family ] [ -c Count ] [ -w timeout ] [ -f | -i Wait ] [ -l Preload ] [ -p Pattern ] [ -s PacketSize ] [ -S hostname/IP addr ] [ -L ] [ -I a.b.c.d. ] [ -o interface ] [ -T ttl ] Host [ PacketSize ] [ Count ]
```

### 説明

`/usr/sbin/ping` コマンドは、インターネット制御メッセージ・プロトコル (ICMP) の ECHO\_REQUEST を送信して、ホストまたはゲートウェイから ICMP ECHO\_RESPONSE を取得します。ping コマンドは、以下の場合に役立ちます。

- ネットワークの状態および各種の外部ホストの判別
- ハードウェアおよびソフトウェアの問題の追跡と分離
- ネットワークの検査、計測、および管理

ホストが操作可能状態であり、ネットワーク上にあれば、エコーに応答します。それぞれのエコー要求は、インターネット・プロトコル (IP) および ICMP ヘッダーを含み、次に ping PID と **timeval** 構造、およびパケットを入れるのに十分なバイト数が続きます。デフォルトでは、割り込みを受信するまで (Ctrl-C)、継続してエコー要求を送信します。

ping コマンドは、1 秒に 1 つのデータグラムを送信し、応答を受信するたびに出力の 1 行を表示します。ping コマンドは、往復時間とパケット損失統計情報を計算し、完了すると簡単に要約して表示します。プログラムがタイムアウトになったとき、または SIGINT シグナルを受信したときに、ping コマンドは完了となります。Host パラメーターは、有効なホスト名か、IP アドレスのいずれかです。

デフォルトでは、ping コマンドは、割り込みを受信するまで (Ctrl-C)、画面表示にエコー要求を送信し続けます。割り込みキーは、stty コマンドを使用して、変更できます。

連続したエコー要求がシステムに及ぼす可能性がある負荷のために、反復要求は問題を分離するために主に使用されます。

### フラグ

<b>-c</b> <i>Count</i>	<i>Count</i> 変数で示されているように、送信 (そして受信) すべきエコー要求数を指定します。このオプションが作用するのは、-c オプションと共に使用した場合のみです。これを使用することで、ping は応答するのに (最後のパケットを送信してから) 「Timeout」の最大秒数まで待ちます。
<b>-w</b> <i>timeout</i>	
<b>-d</b>	ソケット・レベルのデバッグを開始します。
<b>-D</b>	このオプションを使用することで、ICMP ECHO_REPLY パケットの標準出力に 16 進ダンプを出力します。

- f** flood-ping オプションを指定します。**-f** フラグは、パケットが戻ると同じ速さ、または 1 秒あたり 100 回の速さのいずれか速い方に合わせて、「flood」つまり出力します。送信される ECHO\_REQUEST すべてに . (ピリオド) が表示され、受信する ECHO\_REPLY すべてにバックスペース文字が表示されます。これにより、ドロップ中のパケット数を迅速に表示します。root ユーザーだけがこのオプションを使用できます。
- 注: これはネットワーク上では、非常に使いにくい場合がありますので、使用には十分注意してください。flood で ping するには、root ユーザーしか使用できません。
- f** フラグと **-i Wait** フラグには互換性がありません。
- I a.b.c.d** a.b.c.d で指定したインターフェースを発信 IPv4 マルチキャストに使用するように指定します。**-I** フラグは、大文字の i です。
- o interface** interface を発信 IPv6 マルチキャストに使用することを指定します。インターフェースは 'en0'、'tr0' などの形式で指定します。
- i Wait** 各パケットの送信の間では Wait 変数によって指定された秒数だけ待ちます。デフォルトでは、各パケット間で 1 秒待ちます。このオプションは、**-f** フラグと互換性がありません。
- L** マルチキャスト ping のローカル・ループバックを使用不可にします。
- l Preload** Preload 変数で指定したパケット数を通常動作モードに入る前にできるだけ速く (1 秒に 1 つ) 送信します。**-l** フラグは小文字の l です。
- n** 数値出力のみ指定します。ホスト・アドレスのシンボル名を探し出そうとはしません。
- p Pattern** 送信するパケットを埋める「埋め込み」バイトを最大 16 まで指定します。これは、ネットワーク内のデータ依存の問題を診断するのに役立ちます。例えば、**-p ff** は、パケットをすべて 1 のパッドで埋めます。
- q** 抑制出力を指定します。始動時および終了時の要約行以外に、何も表示されません。
- r** 経路指定テーブルをバイパスして、接続されたネットワーク上のホストに直接送信します。Host が直接ネットワークに接続されていない場合、ping コマンドがエラー・メッセージを出します。このオプションを使うと、通過する経路がなくなったインターフェースからローカル・ホストを ping できます。
- R** レコード経路のオプションを指定します。**-R** フラグは、ECHO\_REQUEST パケットに RECORD\_ROUTE オプションを組み込み、戻ったパケットには経路のバッファーを表示します。
- 注: IP ヘッダーには、このような経路は 9 つしか入りません。また、多くのホストおよびゲートウェイがこのオプションを無視します。
- a addr\_family** addr\_family が "inet6" である場合は、ICMP パケットの宛先アドレスを IPv6 フォーマットにマップします。
- s PacketSize** 送信対象のデータのバイト数を指定します。デフォルトは 56 で、8 バイトの ICMP ヘッダー・データと組み合わせられたときには、64 バイトの ICMP データに変換されます。
- S hostname/IP addr** IP アドレスを発信 ping パケット内のソース・アドレスとして使用します。複数の IP アドレスのあるホストでは、**-S** フラグを使用してソース・アドレスを強制的にパケットが送信されるインターフェースの IP アドレス以外にすることができます。IP アドレスがマシンのインターフェース・アドレスの 1 つでない場合は、エラーが戻され、何も送信されません。
- T ttl** マルチキャスト・パケットの生存時間が ttl 秒となるように指定します。
- v** エコー応答ほかに、受信した ICMP パケットをリストする詳細出力を要求します。

## パラメーター

- PacketSize** 送信対象のデータのバイト数を指定します。デフォルトは 56 で、8 バイトの ICMP ヘッダー・データと組み合わせられたときには、64 バイトの ICMP データに変換されます。このパラメーターは、ping コマンドの前のバージョンと互換性を持たせるために組み込まれています。
- Count** 送信 (および受信) 対象のエコー要求数を指定します。このパラメーターは、ping コマンドの前のバージョンと互換性を持たせるために組み込まれています。

## 例

1. ホスト `canopus` とのネットワーク接続を検査して送信するエコー要求数を指定するには、以下のように入力します。

```
ping -c 5 canopus
```

または

```
ping canopus 56 5
```

以下のような情報が表示されます。

```
PING canopus.austin.century.com: (128.116.1.5): 56 data bytes
64 bytes from 128.116.1.5: icmp_seq=0 ttl=255 time=2 ms
64 bytes from 128.116.1.5: icmp_seq=1 ttl=255 time=2 ms
64 bytes from 128.116.1.5: icmp_seq=2 ttl=255 time=3 ms
64 bytes from 128.116.1.5: icmp_seq=3 ttl=255 time=2 ms
64 bytes from 128.116.1.5: icmp_seq=4 ttl=255 time=2 ms
```

```
----canopus.austin.century.com PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 2/2/3 ms
```

2. ホスト `lear` に関する情報を取得して、ソケット・レベルのデバッグを開始するには、以下のように入力します。

```
ping -d lear
```

以下のような情報が表示されます。

```
PING lear.austin.century.com: (128.114.4.18) 56 data bytes
64 bytes from 128.114.4.18: icmp_seq=0 ttl=255 time=6 ms
64 bytes from 128.114.4.18: icmp_seq=1 ttl=255 time=17 ms
64 bytes from 128.114.4.18: icmp_seq=2 ttl=255 time=6 ms
64 bytes from 128.114.4.18: icmp_seq=3 ttl=255 time=6 ms
64 bytes from 128.114.4.18: icmp_seq=4 ttl=255 time=6 ms
^C
```

```
----lear.austin.century.com PING Statistics ----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 6/8/17 ms
```

**注:** 出力は、割り込み (Ctrl-C) を受信するまで繰り返されます。

3. ホスト `opus` についての情報を取得して、送信対象のデータのバイト数を指定するには、以下のように入力します。

```
ping -s 2000 opus
```

または

```
ping opus 2000
```

以下のような情報が表示されます。

```
PING opus.austin.century.com: (129.35.34.234): 2000 data bytes
2008 bytes from 129.35.34.234: icmp_seq=0 ttl=255 time=20 ms
2008 bytes from 129.35.34.234: icmp_seq=1 ttl=255 time=19 ms
2008 bytes from 129.35.34.234: icmp_seq=2 ttl=255 time=20 ms
2008 bytes from 129.35.34.234: icmp_seq=3 ttl=255 time=20 ms
2008 bytes from 129.35.34.234: icmp_seq=4 ttl=255 time=20 ms
2008 bytes from 129.35.34.234: icmp_seq=5 ttl=255 time=19 ms
2008 bytes from 129.35.34.234: icmp_seq=6 ttl=255 time=19 ms
```



```
^C
----opus.austin.century.com PING Statistics----
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 19/19/20 ms
```

注: 出力は、割り込み (Ctrl-C) を受信するまで繰り返されます。

4. ホスト stlopnor に flood-ping オプションを呼び出すには、以下のように入力します。

```
ping -f stlopnor
```

以下のような情報が表示されます。

```
Ping stlopnor.austin.century.com: (129.35.34.234): 56 data bytes
.^C
----stlopnor.austin.century.com PING Statistics ----
1098 packets transmitted, 1097 packets received, 0% packet loss
round-trip min/avg/max = 4/4/11
```

注: flood-ping 出力は、割り込み (Ctrl-C) を受信するまで続きます。

5. ホスト opus に送信されるパケット間の間隔を 5 秒間隔に指定するには、以下のように入力します。

```
ping -i5 opus
```

以下のような情報が表示されます。

```
PING opus.austin.century.com: (129.35.34.234): 56 data bytes
64 bytes from 129.35.34.234: icmp_seq=0 ttl=255 time=5 ms
64 bytes from 129.35.34.234: icmp_seq=1 ttl=255 time=5 ms
64 bytes from 129.35.34.234: icmp_seq=2 ttl=255 time=6 ms
.^C
----opus.austin.century.com PING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 5/5/6 ms
```

注: 出力は、割り込み (Ctrl-C) を受信するまで繰り返されます。

6. *Preload* 変数で指定したパケット数を、通常動作モードに入る前にできるだけ速くホスト opus に送信するには、以下のように入力します。

```
ping -l 10 opus
```

以下のような情報が表示されます。

```
PING opus.austin.century.com: (129.35.34.234): 56 data bytes
64 bytes from 129.35.34.234: icmp_seq=0 ttl=255 time=9 ms
64 bytes from 129.35.34.234: icmp_seq=1 ttl=255 time=11 ms
64 bytes from 129.35.34.234: icmp_seq=2 ttl=255 time=16 ms
64 bytes from 129.35.34.234: icmp_seq=3 ttl=255 time=22 ms
64 bytes from 129.35.34.234: icmp_seq=4 ttl=255 time=26 ms
64 bytes from 129.35.34.234: icmp_seq=5 ttl=255 time=27 ms
64 bytes from 129.35.34.234: icmp_seq=6 ttl=255 time=30 ms
64 bytes from 129.35.34.234: icmp_seq=7 ttl=255 time=31 ms
64 bytes from 129.35.34.234: icmp_seq=8 ttl=255 time=33 ms
64 bytes from 129.35.34.234: icmp_seq=9 ttl=255 time=35 ms
64 bytes from 129.35.34.234: icmp_seq=10 ttl=255 time=36 ms
64 bytes from 129.35.34.234: icmp_seq=11 ttl=255 time=6 ms
64 bytes from 129.35.34.234: icmp_seq=12 ttl=255 time=6 ms
64 bytes from 129.35.34.234: icmp_seq=13 ttl=255 time=6 ms
64 bytes from 129.35.34.234: icmp_seq=14 ttl=255 time=7 ms
64 bytes from 129.35.34.234: icmp_seq=15 ttl=255 time=6 ms
```



```
^C
----opus.austin.century.com PING Statistics----
16 packets transmitted, 16 packets received, 0% packet loss
round-trip min/avg/max = 6/19/36 ms
```

注: 出力は、割り込み (Ctrl-C) を受信するまで繰り返されます。

7. ネットワーク内のデータ依存にかかわる問題を診断するには、以下のように入力します。

```
ping -p ff opus
```

このコマンドは、すべて 1 の埋め込みパターンを持つパケットをホスト `opus` に送信します。以下のような情報が表示されます。

```
PATTERN: 0xff
PING opus.austin.century.com: (129.35.34.234): 56 data bytes
64 bytes from 129.35.34.234: icmp_seq=0 ttl=255 time=5 ms
64 bytes from 129.35.34.234: icmp_seq=1 ttl=255 time=5 ms
64 bytes from 129.35.34.234: icmp_seq=2 ttl=255 time=5 ms
64 bytes from 129.35.34.234: icmp_seq=3 ttl=255 time=6 ms
64 bytes from 129.35.34.234: icmp_seq=4 ttl=255 time=5 ms
^C
----opus.austin.century.com PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 5/5/6 ms
```

注: 出力は、割り込み (Ctrl-C) を受信するまで繰り返されます。

8. 抑制出力を指定するには、以下のように入力します。

```
ping -q bach
```

以下のような要約情報だけが表示されます。

```
PING bach.austin.century.com: (129.35.34.234): 56 data bytes
^C
----bach.austin.century.com PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 5/5/8 ms
```

注: 表示されなくても、パケットの出力は割り込み (Ctrl-C) を受信するまで継続されます。

## 関連情報

`ifconfig` コマンド、`netstat` コマンド。

ネットワークおよびコミュニケーションのマネージの通信およびネットワークのセクション。

---

## pioattred コマンド

### 目的

仮想プリンターの属性のフォーマットと編集の方法を提供します。

### 構文

```
pioattred -q PrintQueueName -d QueueDeviceName [ -o Action] [ -a Attribute]
```

## 説明

**pioattred** コマンドは、仮想プリンターの属性をフォーマットし、属性を編集する方法を提供します。具体的には、**-o** フラグで指定したアクションに従って、プリンターの定義ファイルの属性をフォーマットし、編集すること、またはそのいずれかを行うことができます。フォーマットされた属性は、標準出力 **stdout** に書き込まれます。**VISUAL** 環境変数に指定したエディターを使って属性を編集します。仮想プリンターの定義ファイルは、**/var/spool/lpd/pio/@local/custom/\*** ディレクトリーにあるものと想定されます。

## フラグ

- a Attribute** 仮想プリンターの定義ファイルの属性名を指定して、フォーマットまたは編集を行います。このフラグは、何度も指定できます。
- d QueueDeviceName** 仮想プリンター定義の *QueueDeviceName* スプーラーを指定して、フォーマットまたは編集を行います。
- o Action** **pioattred** コマンドが仮想プリンター定義で取るべきアクションを指定します。このフラグを省略した場合、**pioattred** コマンドは 0 (ゼロ) の値を想定します。
- 0** 指定された属性をフォーマットします。その結果は **stdout** に行きます。
  - 1** 指定された属性の 1 つまたは複数フォーマットし、編集します。**VISUAL** 環境変数に指定したエディターを使用します。**VISUAL** 環境変数にエディターが指定されていない場合、**vi** エディターを使用します。属性を編集中にエラーが起きた場合には、そのエラーのある属性を一時ファイルに保管し、エラーを示す戻りコードを戻します。
- 以下の値は、属性を編集した後で、エラー戻りコードが戻った場合に使用されません。
- 2** 属性を再編集します。仮想プリンターの定義は、エラーが起きたときの状態になります。
  - 3** エラーを無視し、編集した属性を仮想プリンターの定義に保管します。
  - 4** クリーンアップして、**pioattred** コマンドの開始前の状態にしておきます。
- q PrintQueueName** 仮想プリンター定義の *PrintQueueName* スプーラーを指定して、フォーマットまたは編集を行います。

## 例

- キュー *quedev* 仮想プリンターの定義に **ci** および **sh** 属性をフォーマットするには、以下のように入力します。

```
pioattred -q queue -d quedev -o 0 -a ci -a sh
```

または

```
pioattred -q queue -d quedev -a ci -a sh
```

- キュー *quedev* 仮想プリンターの定義にすべての属性をフォーマットするには、以下のように入力します。

```
pioattred -q queue -d quedev -o 0
```

または

```
pioattred -q queue -d quedev
```

3. キュー `quedev` 仮想プリンターの定義で `st` 属性を編集するには、以下のように入力します。

```
pioattred -q queue -d quedev -o 1 -a st
```

## ファイル

`/usr/sbin/pioattred` `pioattred` コマンドが入っています。

## 関連情報

`lsvirprt` コマンド。

プリンターおよび印刷 の 仮想プリンターの定義と属性 のセクション。

---

## piobe コマンド

### 目的

プリンター・バックエンドの印刷ジョブ・マネージャーです。

### 構文

```
/usr/lpd/piobe [ -a PreviewLevel ] [ -A DiagnosticLevel ] [ -d InputDataStream ] [ -f FilterName ]  
[ FormatterFlags ] [ File ... ]
```

### 説明

`piobe` コマンドは、印刷ジョブを処理するために `qdaemon` プログラムによって呼び出されるスプーラー・バックエンド・プログラムです。 `piobe` コマンドは、印刷ジョブ・マネージャーとして機能します。

印刷ファイルのデータ・ストリーム・タイプを指定する `-d` フラグの引数 (またはそのデータベース内のデフォルト値) に基づいて、`piobe` コマンドは、データベースからパイプラインを取り出して、シェルに渡します。パイプラインには、必要に応じて印刷ファイルを変換してプリンターに送信するフィルターの文字列が入っています。 要求すると、`piobe` コマンドは、データベースからパイプラインを取り出して実行して、ヘッダー・ページおよびトレーラー・ページも生成します。

`FormatterFlags` 引数 (この項でリストする以外のフラグ) は、パイプライン内でフィルター・コマンドによって参照されるものと想定されます。フラグを指定してもパイプラインによって参照されないと、エラー・メッセージが発行されて印刷ジョブが終了します。

**注:** コマンド行に直接 `piobe` コマンドを入力しないでください。このコマンドは、`qdaemon` プロセスで呼び出され、`qdaemon` プロセスが提供する各種サービスに従います。

## フラグ

### -a PreviewOption

実際にファイルを印刷せずに、印刷ジョブのパラメーター値を事前に確認する方法を提供します。 *PreviewOption* 変数には、以下の値を指定できます。

- 0** 通常の印刷処理を指定します。
- 1** 入力データ・タイプをプリンターで予想されるデータ・タイプに変換するときに使うフラグ値とフィルターのパイプラインのリストを戻します。ただし、実際には、フィルターのパイプラインを呼び出したり、ファイルをプリンターに送信したりしません。

戻ったフラグ値のリストは、構成データベースからのデフォルト時のコマンド行フラグ値です。これらの値は、コマンド行に指定したフラグの引数によって無効になります。以下の点に注意してください。

- **-d** フラグで指定した (または、デフォルトの) *InputDataType* 変数に対して有効なフラグだけが表示されます。
- 実際の印刷ではなく印刷ジョブのスプーリングにのみ関連するフラグ値は表示されません。スプーリング・フラグのデフォルト値は、**qprt** コマンドのフラグの説明とともに組み込まれています。
- フラグ値は、有効かどうか検査されていない場合があります。

フィルターのパイプラインは、印刷ファイルからデータを処理するフィルター・コマンド (およびフィルター・コマンドに渡されるフラグ値) を表示してから、プリンターに送信します。各フィルター・コマンドの記述を検討して、実行されるフィルター操作のタイプを判別できます。

### -A Value

診断出力のレベルを指定します。診断出力は、印刷ファイル、ヘッダー・ページ、またはトレーラー・ページを処理しているフィルターのパイプラインによって見つけられたエラーの診断に役立ちます。診断出力は印刷ジョブを実行要求したユーザーにメールされます。 *Value* 変数には、以下のいずれかを使うことができます。

- 0** ヘッダー、トレーラー、または印刷ファイルのパイプラインによって生成される標準エラー出力を破棄します。
- 1** 標準エラー出力が生成されると、標準エラー出力およびそれを生成したパイプラインを戻して、印刷ジョブを終了します。
- 2** フラグ値、標準エラー出力 (あれば) を戻し、エラーを検出したかどうかに関係なく、パイプラインを完了します。エラーを検出した場合には、印刷ジョブが終了します。
- 3** 値が **2** の場合と似ていますが、ファイルは印刷されません。

望ましい値は **1** です。パイプライン内のフィルターが標準エラーに出力されると、状況情報などに関するエラーがなくても値 **0** が使用されます。問題が発生したときに標準エラー出力が生成されない場合でも、問題を診断するには **2** か **3** が使用されます。

**-d** *InputDataType*

印刷対象のファイルにあるデータ・タイプを指定します。このフラグは 1 文字の ID です。印刷ファイルのデータ・タイプとプリンターが予想するデータ・タイプに基づいて、プリンターに送信される前に印刷ファイルが (必要であれば) フィルターを通して渡されます。データ・タイプ ID の例は以下のとおりです。

- a** IBM 拡張 ASCII
- p** パススルー (変更されずにプリンターに送信されます)
- s** PostScript
- c** PCL
- d** Diablo 630
- k** 漢字

ユーザーの選択したプリンターが *InputDataType* 変数をサポートせずに、かつユーザーの印刷ファイルのデータ・タイプをプリンターがサポートするデータ・タイプに変換するのにフィルターを使用できない場合、印刷ジョブは、エラー・メッセージを表示して終了します。

**-f** *FilterType*

ユーザーの印刷ファイルをプリンターへ送信する前に通過させるフィルターのタイプを指定します。このフラグは 1 文字の ID です。この ID は、**lpr** コマンドで使用できるフィルター・フラグに類似しています。使用できるフィルター ID は **p** で、これは **pr** フィルターを呼び出します。

## ファイル

`/var/spool/lpd/pio/@local/ddi/*`

要約されたデータベース・ファイルが入っています。

## 関連情報

**lpr** コマンド、**pr** コマンド、**qdaemon** コマンド、**qprt** コマンド。

**qconfig** ファイル。

プリンターおよび印刷 の プリンター・バックエンド・プログラミング のセクション。

プリンターおよび印刷 の バックエンドおよび qdaemon の相互作用のセクション。

プリンターおよび印刷 の libqb 内のバックエンド・ルーチン のセクション。

「*AIX 5L Version 5.3 Kernel Extensions and Device Support Programming Concepts*」の Printer Addition Management Subsystem: Programming Overview のセクション。

プリンターおよび印刷 の キューを追加しない場合のプリンターの構成 (SMIT) のセクション。

プリンターおよび印刷 の プリンター・コロンのファイルを使用したプリンターの追加 のセクション。

---

## pioburst コマンド

### 目的

プリンター出力のバースト・ページ (ヘッダー・ページおよびトレーラー・ページ) を生成します。

## 構文

`/usr/lpd/pio/etc/pioburst [ -H HostName ] TextFile`

## 説明

**pioburst** コマンドは、*TextFile* 変数で指定したファイルからバースト・ページのプロトタイプ・テキストを取り出し、そのプロトタイプ・テキストの % エスケープ・シーケンスで識別された変数フィールドに入れて、構築したテキストを標準出力に書き出します。これは、印刷ジョブ・マネージャーの **piobe** コマンドによってパイプラインのフィルターとして呼び出されます。

対応する値で置換される % エスケープ・シーケンスは、以下のとおりです。

**%A** フォーマットしているフラグ値を指定します。  
**%D** 印刷の出力先ユーザーを指定します。  
**%H** ジョブを印刷しているホスト・コンピューター名を指定します。  
**%P** 印刷ジョブが印刷される時刻を指定します。  
**%Q** 印刷ジョブがキューに入れられた時刻を指定します。  
**%S** 印刷ジョブを実行依頼したユーザーを指定します。  
**%T** 印刷ジョブの表題を指定します。  
**%%** % (パーセント記号) を指定します。

それぞれの変数フィールドごとのラベル (長さ 20 文字) は、小文字を使用する以外は、変数フィールドと同じエスケープ・シーケンスを使用して指定できます。例えば、印刷ジョブがキューに入れられたこと (**%Q**) を指定する変数フィールドのラベルを生成するには、**%q** を使用します。**%e** 変数は、ラベル **END OF OUTPUT FOR:** を表します。

**pioburst** コマンドでは、以下の環境変数を初期化する必要があります。

**PIOTITLE** 印刷ジョブの表題 (**%T**)  
**PIOQDATE** 印刷ジョブがキューに入れられた時刻 (**%Q**)  
**PIOFROM** 印刷ジョブを実行依頼したユーザー (**%S**)  
**PIOTO** 印刷の出力先ユーザー (**%D**)  
**PIOFLAGS** フラグ値 (**%A**)

## フラグ

**-H *HostName*** *HostName* 変数で指定されたホスト名がデフォルトのホスト名 (ジョブを印刷するホスト・コンピューター名) を無効にするように指定します。

## 例

ヘッダー・ページを生成し、標準出力に送信するには、以下のように入力します。

```
pioburst /usr/lpd/pio/burst/H.ascii
```

## ファイル

`/usr/lpd/pio/etc/pioburst`

**pioburst** コマンドが入っています。

## 関連情報

**digest** コマンド、**piobe** コマンド、**piodigest** コマンド、**pioformat** コマンド、**piofquote** コマンド、**pioout** コマンド、**pioprepend** コマンド。

プリンターおよび印刷 の 印刷の管理 のセクション。

プリンターおよび印刷 の プリンター・コロンのファイルの規則 のセクション。

プリンターおよび印刷 の プリンター・コロンのファイルのエスケープ・シーケンス のセクション。

プリンターおよび印刷 の プリンター・コード・ページ変換テーブル のセクション。

プリンターおよび印刷 の 仮想プリンターの定義と属性 のセクション。

---

## piocnvt コマンド

### 目的

事前定義プリンターの定義または仮想プリンターの定義を拡張または縮小します。

### 構文

```
piocnvt [ -s State ] -i SourceFile [ -o TargetFile ]
```

### 説明

**piocnvt** コマンドは、事前定義プリンターの定義あるいは仮想プリンターの定義のいずれかを取り、ファイルを拡張または縮小します。拡張されたプリンター定義ファイルには、そのプリンター定義に対応付けられたすべての属性が含まれます。縮小されたプリンター定義には、そのプリンター定義に固有のプリンター属性だけが含まれます。

プリンターの定義ファイルは、階層構造の親子関係で配列されています。例えば、事前定義プリンターの定義 `4201-3.asc` は、親 `master` を持ちます。`4201-3.asc` の拡張プリンターの定義は、`master` からの属性と `4201-3.asc` からの属性をすべて含むこととなります。`4201-3.asc` の縮小プリンターの定義は、`master` にない属性だけが含まれることとなります。**piocnvt** コマンドは、単にプリンターの定義ファイルの拡張状況と縮小状況の間を往復する方法を提供するだけです。

### フラグ

- i SourceFile** 完全パスおよび入力ファイル名を指定します。
- o TargetFile** 完全パスおよびその出力ファイル名を指定します。**-o** フラグを省略すると、出力に *SourceFile* が使われます。
- s State** *TargetFile* パラメーターの状態を拡張するか縮小するかを指定します。**-s** フラグを省略すると、**piocnvt** コマンドが *SourceFile* の **zD** 属性を検査することによってその状態を判別します。判別できない場合、*TargetFile* パラメーターが拡張状態におかれます。
  - +** *TargetFile* パラメーターの状態が拡張されることを示します。
  - !** *TargetFile* パラメーターが縮小されることを示します。



## 例

1. 仮想プリンターの定義 lp0:lp0 をファイル new:lp0 に拡張するには、以下のように入力します。  

```
piocnvt -s+ -i lp0:lp0 -o new:lp0
```
2. 仮想プリンターの定義 lp0:lp0 を適度に縮小するには、以下のように入力します。  

```
piocnvt -s! -i lp0:lp0
```

## ファイル

`/usr/sbin/piocnvt` `piocnvt` コマンドが入っています。

## 関連情報

`chvirprt` コマンド、`lsvirprt` コマンド、`mkvirprt` コマンド。

プリンターおよび印刷 の 印刷の管理 のセクション。

プリンターおよび印刷 の 印刷スプーラー のセクション。

プリンターおよび印刷 の プリンター固有の情報 のセクション。

プリンターおよび印刷 の 追加したプリンターのサポートのインストール のセクション。

プリンターおよび印刷 の 仮想プリンターの定義と属性 のセクション。

---

## piodigest コマンド

### 目的

仮想プリンターの定義の属性値を整理してメモリー・イメージに入れ、メモリー・イメージをファイルに格納します。

### 構文

```
/usr/lpd/pio/etc/piodigest [ -s DataStreamType ] [ -n DeviceName ] [ -p DirectoryPath ] [ -q PrintQueueName ] [ -t PrinterType ] [ -d QueueDeviceName ] { ColonFileName | - }
```

### 説明

`piodigest` コマンドは、コロン・ファイルから仮想プリンターの属性値を取り出し、取り出した属性値およびその索引テーブルのメモリー・イメージを構築し、その構築したメモリー・イメージをファイルに書き込みます。ファイルにあるメモリー・イメージは、プリンター・フォーマッターおよび `piobe` コマンド (印刷ジョブ・マネージャー) によって、アクセスする場合に使用できます。また、このコマンドは、指定されたキューとキュー・デバイスのためのオブジェクト・データ・マネージャー (ODM) スタンザを作成します。ODM スタンザは、System Management Interface Tool (SMIT) のダイアログで使用します。`zV` と呼ばれる属性が指定され、かつその属性に `+` の値が入っている場合は、`piodigest` コマンドは、コロン・ファイルに指定されたすべての属性に対して構文、参照、および制限の妥当性検査を行います。

仮想プリンターの定義バージョンを初めてカスタマイズするときか、後で変更するときには必ず、

`piodigest` コマンドを呼び出してください。`piodigest` コマンドを呼び出すたびに、1 つの仮想プリンターの定義について属性値を要約します。

*ColonFileName* パラメーターは、コロン・フォーマットでの入力ファイルの名前です。コロン・ファイルには、1 つの仮想プリンターの属性値が入っています。 *ColonFileName* パラメーターに対する - (ダッシュ) の値は、標準入力からコロン・ファイルを読み取るべきであることを示しています。

生成される出力ファイル名は、以下の書式になります。

PrinterType.DataStreamType.DeviceName.PrintQueueName:QueueDeviceName

## フラグ

- d** *QueueDeviceName* 仮想プリンター (キュー・デバイス) の名前を指定します。
- このフラグを指定しない場合、入力コロン・ファイルの **mv** 属性によって指定された仮想プリンター名が想定されます。
- n** *DeviceName* プリンターの名前として、ライン・プリンター 0 の場合は lp0 と指定し、ライン・プリンター 1 の場合は lp1 と指定します。
- このフラグを指定しない場合、入力コロン・ファイルの **mn** 属性で指定されたデバイス名が想定されます。
- p** *DirectoryPath* 出力ファイルが生成されるディレクトリーのパス名を指定します。
- このフラグを指定しない場合、**/var/spool/lpd/pio/@local/ddi** ディレクトリーが想定されます。
- q** *PrintQueueName* 仮想プリンターが割り当てられている印刷キュー名を指定します。
- このフラグを指定しない場合、入力コロン・ファイルの **mq** 属性によって指定される印刷キュー名が想定されます。
- s** *DataStreamType* プリンターのデータ・ストリーム・タイプを指定します。サンプルのデータ・ストリーム・タイプは、**asc** (IBM 拡張 ASCII)、**ps** (PostScript)、**pcl** (HP PCL)、および **630** (Diablo 630) です。
- このフラグを指定しない場合、入力コロン・ファイルの **md** 属性によって指定されたデータ・ストリーム・タイプが想定されます。
- t** *PrinterType* プリンターのタイプを指定します。4201-3 や ti2115 などです。
- このフラグを指定しない場合、入力コロン・ファイルの **mt** 属性によって指定されたプリンターのタイプが想定されます。

## 例

要約された仮想プリンター定義を生成するには、以下のように入力します。

```
piodigest -d mypro -n lp0 -q proq -s asc -t 4201-3
```

proq 印刷キューの mypro キュー・デバイスに割り当てられた仮想プリンターの属性値は、要約されて、**/var/spool/lpd/pio/@local/ddi** ディレクトリーの **4201-3.asc.lp0.proq:mypro** という名前のファイルに格納されます。

## ファイル

**/var/spool/lpd/pio/@local/ddi/\*** 要約された仮想プリンターの定義が入っています。  
**/usr/lpd/pio/etc/piodigest** piodigest コマンドが入っています。

## 関連情報

**mkvirprt** コマンド、**piobe** コマンド、**pioburst** コマンド、**pioformat** コマンド、**piofquote** コマンド、**pioout** コマンド、**piopredef** コマンド。

プリンターおよび印刷の印刷の管理 および 仮想プリンターの定義と属性、プリンター・コード・ページ変換テーブル、プリンター・コロンのファイルの規則、プリンター・コロンのファイルのエスケープ・シーケンス、プリンター・バックエンドのプログラム、バックエンドと `qdaemon` の相互作用、`libqb` 内のバックエンド・ルーチンのセクション。

---

## piodmgr コマンド

### 目的

`/var/spool/lpd/pio/@local/smit` ディレクトリー内のオブジェクト・データ・マネージャー (ODM) のデータベースを圧縮します。

### 構文

```
piodmgr { -c | -h }
```

### 説明

**piodmgr** コマンドは、`/var/spool/lpd/pio/@local/smit` ディレクトリー内の ODM データベースから既存のプリンターの定義を取り出し、ODM データベースを再作成し、データベースを圧縮し、圧縮したデータベースを再ロードします。

フラグの **-c** および **-h** は、相互に排他的です。**-h** フラグを指定すると、ホスト名が変更されているときにのみデータベースが圧縮されます。**-c** フラグを指定すると、データベースが常に圧縮されます。

注: このコマンドを実行するには、`root` ユーザー権限が必要です。

### フラグ

- c** ODM データベースから既存のプリンターの定義を取り出し、データベースを再作成し、情報を圧縮して、データベースを置換します。
- h** **-c** フラグを指定した場合とまったく同じに機能しますが、**-h** フラグを指定すると、ホスト名が変更されている場合にのみ情報が圧縮されます。ホスト名が変更されている場合、**-h** フラグを指定すると、新しい名前が取り出されて、データベースのホスト名情報が更新されます。ホスト名が変更されていない場合、**-h** フラグを指定しても、情報が圧縮されません。このフラグは、**-c** フラグのような自動圧縮フラグではなく、オプションの圧縮フラグです。

### 例

1. ODM プリンター定義データベースを圧縮し、更新するには、以下のように入力します。

```
piodmgr -c
```
2. ホスト名が変更されているかどうかに応じて情報を圧縮するには、以下のように入力します。

```
piodmgr -h
```

## ファイル

`/usr/lib/lpd/pio/etc/piodmgr` `piodmgr` コマンドが入っています。  
`/var/spool/lpd/pio/@local/smit/*` コマンドによって使用される事前に定義されたプリンター定義が入っています。

## 関連情報

`piobe` コマンド。

プリンターおよび印刷 の 印刷の管理、印刷スプーラー、および プリンター・バックエンドのプログラムのセクション。

AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグのプログラマーのためのオブジェクト・データ・マネージャー (ODM) の概要のセクション。

---

## piofontin コマンド

### 目的

マルチリンガル・フォントのディスケットからフォントをコピーします。

### 構文

`piofontin -t PrinterType -c Codepage [ -d Device ]`

### 説明

`piofontin` コマンドは、マルチリンガル・フォントのディスケットから、`/usr/lib/lpd/pio/fonts` ラベルより 1 レベル下のディレクトリーにフォント・ファイルをコピーします。フォント・ファイルがコピーされるディレクトリーは `PrinterType` パラメーターで指定した名前になります。フォント・ファイルには、ファイルの命名規則に従って名前が付けられます。名前は、以下のフォーマットになります。

`codepage.typeface.pitch*10.quality`

root ユーザーだけが `piofontin` コマンドを使用できます。

### フラグ

- `-c Codepage` フォントについてコード・ページを指定します。ギリシャ語のフォントの場合、この値は 851 であり、トルコ語のフォントでは、853 になります。
- `-d Device` ディスケット・ドライブのデバイス名を指定します。このデフォルトは、`-d/dev/fd0` ラベルで、3.5 インチの標準ディスケット・ドライブです。
- `-t PrinterType` フォントについてプリンターのタイプを指定します。サポートされるプリンターのタイプは、4201-3、4202-3、4207-2、4208-2、2380、2381、2390、および 2391 です。

### 例

ディスケット・ドライブ `/dev/fd1`; からコード・ページ 851 の 4201-3 フォントが入っているディスケットを読み取るには、以下のように入力します。

```
piofontin 4201-3 851 /dev/fd1
```

フォント・ファイルは `/usr/lib/lpd/pio/fonts/4201-3` ディレクトリーにコピーされます。

## ファイル

`/usr/sbin/piofontin`

`piofontin` コマンドが入っています。

## 関連情報

プリンターおよび印刷の印刷の管理、プリンター固有の情報、追加したプリンターのサポートのインストール、仮想プリンターの定義と属性、プリンター・コード・ページ変換テーブルのセクション。

---

## pioformat コマンド

### 目的

プリンター・フォーマッターを始動します。

### 構文

```
/usr/lpd/pio/etc/pioformat -@ DataBaseFile [ -! FormatterName ] [ -# + PassThroughOption ]
```

### 説明

`pioformat` コマンドは、プリンター・フォーマッター・ドライバーを開始します。フォーマッター・ドライバーは、データベース値へのアクセスを設定し、プリンター・フォーマッターをロードしてリンクし、次にフォーマッターの `setup` 関数、`initialize` 関数、`lineout` 関数、`passthru` 関数、および `restore` 関数を適宜呼び出すことでフォーマッターを始動します。フォーマッター・ドライバーは、フォーマッターが使用する `piogetvals` サブルーチン、`piogetopt` サブルーチン、`piocmdout` サブルーチン、`piogetstr` サブルーチン、`piomsgout` サブルーチン、および `pioexit` サブルーチンも提供します。

以下のフラグは、フォーマッター・ドライバーが処理し、フォーマッターに渡されることはありません。ただし、以下にリストされていないすべてのフラグがフォーマット化フラグとして想定され、フォーマッターへ渡されます。

### フラグ

`-@ DataBaseFile`

以下のいずれかを指定します。

- アクセスする (要約された) データベース・ファイルの絶対パス名。
- コロンで区切られた印刷キューおよびキュー・デバイス名。

引数文字列が `/` (スラッシュ) 文字で始まる場合、絶対パス名であると想定されます。

キューの名前とキュー・デバイス名の組み合わせると、データベース・ファイル名の一部である固有文字列になり、`/var/spool/lpd/pio/@local/ddi` ディレクトリ内でデータベース・ファイル名の検索に使用されます。この簡易書式の代替名は、フォーマッター・ドライバーとフォーマッターがスタンドアロン・デバイスとして実行されるときに、スプーラーの代わりに、その便宜を図るものとして与えられます。

**-!** *FormatterName*

ロード、リンク、および始動の対象となるフォーマッターの絶対パス名を指定します。

**-!** フラグを指定しない場合、データベースの **mf** 属性名で定義されたデフォルトのフォーマッター名が使用されます。デフォルト・フォーマッター名は、フォーマッター・ドライバーとフォーマッターがスタンドアロン・デバイスとして実行されるときに、スプーラーの代わりに、その便宜を図るものとして与えられます。

**-# +** *PassThroughOption*

印刷ファイルは変更されずにパススルーされるものとして指定します。 **-# +** フラグを指定しない場合、印刷ファイルがフォーマットされます。

フォーマッターの **setup** ルーチンに渡されるパラメーターには、値 **0** ではなく、**1** が入っています。値 **0** は、ファイルをフォーマットする代わりにパススルーすることを示します。

## 例

1. ページ幅を 132 文字にオーバーライドし、**pioformat** コマンドとフォーマッターをスタンドアロン・フィルターとして使用して、印刷キュー **pro** に対応するキュー・デバイス **std** について、データベース・ファイル (仮想プリンター記述) にしたがって、**myfile** ファイルをフォーマットするには、以下のように入力します。

```
cat myfile | pioformat -@ pro:std -w 132 >/dev/lp0
```

2. スプーラーのもとで実行中のパイプライン内で **pioformat** コマンドとフォーマッターを使用するには、以下のように入力します。

```
%Ide/pioformat -@ %Idd/%Imm -! %Idf/piof420x %Fbb %Fee ...
```

この例では、以下のことが想定されています。

- プリンターは、4207 Model 2 Proprinter です。
- 印刷キュー名は **pro** です。
- 印刷キューについて定義されているキュー・デバイス (仮想プリンター) は、**std** というキュー 1 つだけで、その出力データ・ストリーム・タイプは **asc** (拡張 ASCII) です。
- プリンター名は **/dev/lp0** です。
- 印刷ジョブ実行依頼側がフラグと引数 **-i 5** を指定しています。

印刷ジョブ・マネージャー (**piobe** コマンド) がシェルにパイプラインを渡してファイルをフォーマット設定する前に、属性値に対するパイプラインの組み込み参照を解決します。この例についての上記の想定に基づき、属性参照が以下のように解決されます。

```
%Ide -> /usr/lpd/pio/etc
```

**pioformat** コマンドが常駐するディレクトリー。

```
%Idd ->/var/spool/lpd/pio@local/ddi
```

データベース・ファイルのディレクトリー。

```
%Imm ->4207-2.asc.lp0.pro:std
```

データベース・ファイル名。

```
%Idf -> /usr/lpd/pio/fmtrs
```

フォーマッターのディレクトリー。

```
%Fbb ->
```

空文字列 (実行依頼側が **-b** フラグを指定しなかったため)

%Fee -> -i 5

このフラグと引数を指定した実行依頼。

結果的に生成された以下のパイプラインがシェルに渡されてファイルをフォーマットします (読みやすいように複数行に分けて掲載してあります)。

```
/usr/lpd/pio/etc/pioformat      # initiate the formatter driver
-@/usr/lpd/pio/ddi/4207-2.asc.lp0.pro:std
                                # (digested) database file
-!/usr/lpd/pio/fmtrs/piof420x  # loadable formatter
-i5                             # formatting option
                                # (indent 5 characters)
```

## ファイル

<code>/usr/lpd/pio/etc/pioformat</code>	フォーマッター・ドライバーが入っています。
<code>/usr/lpd/pio/fmtrs/*</code>	フォーマッターが入っています。
<code>/var/spool/lpd/pio/@local/ddi/*</code>	要約されたデータベース・ファイルが入っています。

## 関連情報

**pio**be コマンド、**pioburst** コマンド、**pidigest** コマンド、**pioquote** コマンド、**pioout** コマンド、**pioprepend** コマンド。

**piocmdout** サブルーチン、**piogetvals** サブルーチン、**piogetopt** サブルーチン、**piogetstr** サブルーチン、**piomsgout** サブルーチン、**pioexit** サブルーチン。

プリンターおよび印刷 の 印刷の管理 のセクション。

プリンターおよび印刷 の 仮想プリンターの定義と属性 のセクション。

「*AIX 5L Version 5.3 Kernel Extensions and Device Support Programming Concepts*」の Printer Addition Management Subsystem: Programming Overview のセクション。

プリンターおよび印刷 の プリンター・コード・ページ変換テーブル のセクション。

プリンターおよび印刷 の プリンター・コロンのファイルの規則 のセクション。

プリンターおよび印刷 の プリンター・コロンのファイルのエスケープ・シーケンス のセクション。

プリンターおよび印刷 の プリンター固有の情報 のセクション。

プリンターおよび印刷 の 追加したプリンターのサポートのインストール のセクション。

プリンターおよび印刷 の プリンター・バックエンドのプログラム のセクション。

プリンターおよび印刷 の プリンター・コロンのファイルを使用したプリンターの追加 のセクション。

---

## pioquote コマンド

### 目的

PostScript プリンター用の特定の制御文字を変換します。



## 構文

`/usr/lpd/pio/etc/piofquote`

## 説明

**piofquote** コマンドは、他のプリンターをエミュレートできる PostScript プリンターを宛先とした特定の制御文字を変換するフィルターです。このコマンドは、標準入力からデータを読み取り、制御文字について検査し、それらを適宜変更します。次にそのデータを標準出力へ書き出します。

少なくとも 1 バイトのデータが標準入力に表示されれば、最初の入力データのバイトが標準出力に書き出される前に **piofquote** コマンドが 16 進の 04 制御文字を標準出力に書き出します。このコマンドは、ファイルの終わりを標準入力で認知したときにも、標準出力に 16 進の 04 を書き込みます。

標準入力から読み取った入力データに 16 進の 01、03、04、05、11、13、14、または 1c の制御文字がある場合、その制御文字の 16 進の 40 ビットがオンになり、標準出力に書き出される前に 16 進の 01 文字が接頭部として制御文字に書き込まれます。

## ファイル

<b>standard input</b>	処理される入力データ・ストリーム。
<b>standard output</b>	変換された制御文字が入っている出力データ・ストリーム。

## 関連情報

**pio** コマンド、**pioburst** コマンド、**piodigest** コマンド、**pioformat** コマンド、**pioout** コマンド、**pioprepend** コマンド。

プリンターおよび印刷 の 印刷の管理 のセクション。

---

## piolsvp コマンド

### 目的

システム上の仮想プリンターをリストします。

### 構文

```
piolsvp { -q | -v | -Q | -p | -A } [ -nAttachmentField]
```

```
piolsvp -P Queue [ : QueueDevice] -nAttachmentField
```

```
piolsvp -P Queue -d
```

```
piolsvp -N AttachmentType -nAttachmentField
```

### 説明

**piolsvp** コマンドは、システム上の仮想プリンターと接続タイプをリストします。**piolsvp** コマンドは、仮想プリンターへのキュー、またはキューとキュー・デバイスの組のいずれかを表示します。

キューとキュー・デバイスの組のリスト順は、`/etc/qconfig` ファイルに用いられる順序と同じです。

## フラグ

- A** すべての接続タイプとその説明を表示します。 `/usr/lib/lpd/pio/etc` ディレクトリーにある `.attach` と `.config` ファイルにより、すべての接続タイプが定義されます。
- d** 指定されたキューに関連したキュー・デバイスを表示します。
- nAttachmentField** 接続のためのフィールド名を指定します。一般に、フィールド名は SMIT セレクター名です。 `AttachmentField` 変数に使用できる値は、以下のとおりです。
- submit\_job**
- add\_queue**
- add\_printer**
- remove\_queue**
- printer\_conn**
- change\_queue**
- change\_filters**
- n** や **-A** フラグが指定されると、接続ファイル内の指定された接続フィールドの値を持つ接続タイプのみが表示されます。接続定義は、`AttachmentType.attach` 命名規則に従って、ファイルに保持されます。 `.attach` ファイルは、`/usr/lib/lpd/pio/etc` ディレクトリーに常駐します。
- n** フラグを **-q** または **-v** フラグのいずれかと一緒に指定すると、定義された接続タイプに属するキューとキュー・デバイスの組だけが表示されます。定義済みの接続タイプは、定義ファイル内でフィールド値が割り当てられています。
- n** フラグを **-P** フラグとともに指定すると、SMIT セレクター名が表示されます。**-n** と **-P** フラグの組み合わせを指定すると、キュー・デバイス名と接続タイプも表示されます。
- n** フラグを **-N** フラグとともに指定すると、指定した接続フィールドと接続タイプに関する SMIT セレクター名が表示されます。
- N** 接続タイプを指定します。所定の接続フィールドに関連付けられた SMIT セレクター名が表示されます。
- p** システム上にキューとキュー・デバイスのすべての組を表示して、それぞれの組ごとに説明します。複数のキュー・デバイスを持つキューがあれば、最初のキュー・デバイスのキューの名前のみが表示されます。
- P** 情報を表示するキューの名前またはキュー・デバイス名を指定します。情報は、キュー・デバイス名、接続タイプ名、および SMIT セレクター値名で構成されます。
- q** システム上のすべてのキューが表示されます。 **-q** フラグを指定すると、複数のデバイスを持つキューのキュー・デバイスの組も表示されます。
- Q** システム上のすべてのキューが表示されます。 **-Q** フラグを指定しても、キュー・デバイスの組は表示されません。 **-q** フラグを指定して、キュー・デバイスの組をリストします。
- v** 仮想プリンターを持つキューのキュー・デバイスのすべての組が表示されます。

## 例

1. システム上に印刷キューをすべて表示するには、以下のように入力します。

```
piolsvp -q
```

このコマンドの出力は以下のようになります。

```
e4019a          4019 (IBM ASCII)
d3816          IBM 3816 Page Printer
ena_asc        4029 (IBM ASCII)
ena_gl         4029 (Plotter Emulation)
ena_pcl        4029 (HP LaserJet II Emulation)
ena_ps         4029 (PostScript)
hplj2          Hewlett-Packard LaserJet II
tstx           4216-31 (Proprinter XL Emulation)
e4019ps        4019 (PostScript)
4019lxxa       4029 (PostScript)
4019lxxa:lxx   4029 (PostScript)
4019lxxa:rkm1xx 4019 (IBM ASCII)
4019lxxa:rkm1xxl 4019 (IBM ASCII)
```

2. システム内のすべての仮想プリンターを表示するには、以下のように入力します。

```
pio1svp -v
```

このコマンドの出力は以下のようになります。

```
#QUEUE          DEVICE          DESCRIPTION
e4019a          e4019           4019 (IBM ASCII)
d3816           ena3816         IBM 3816 Page Printer
ena_asc         ena              4029 (IBM ASCII)
ena_gl          ena              4029 (Plotter Emulation)
ena_pcl         ena              4029 (HP LaserJet II Emulation)
ena_ps          ena              4029 (PostScript)
hplj2           lxx              Hewlett-Packard LaserJet II
tstx            lxx              4216-31 (Proprinter XL Emulation)
e4019ps         e4019           4019 (PostScript)
4019lxxa        lxx              4029 (PostScript)
4019lxxa        rkm1xx          4019 (IBM ASCII)
4019lxxa        rkm1xx          4019 (IBM ASCII)
```

3. システム上のすべてのキューをリストするには、以下のように入力します。

```
pio1svp -Q
```

このコマンドの出力は以下のようになります。

```
e4019a          4019 (IBM ASCII)
d3816          IBM 3816 Page Printer
ena_asc        4029 (IBM ASCII)
ena_gl         4029 (Plotter Emulation)
ena_pcl        4029 (HP LaserJet II Emulation)
ena_ps         4019 (PostScript)
hplj2          Hewlett-Packard LaserJet II
tstx           4216-31 (Proprinter XL Emulation)
e4019ps        4019 (PostScript)
4019lxxa       4029 (PostScript)
```

4. SMIT セレクター `add_queue` に値が指定されている接続タイプすべてをリストするには、以下のように入力します。

```
pio1svp -A -nadd_queue
```

このコマンドの出力は以下のようになります。

```
#ATTACHMENT TYPE    DESCRIPTION
local               Local Attached
remote              Remote Attached
ascii                ASCII Terminal Attached
other                Generic Backend Attached
```

5. キュー `4019lxxa` に関する情報をリストするには、以下のように入力します。

```
pio1svp -P4019lxxa -n add_queue
```

このコマンドの出力は以下のようになります。

```
lxx      xsta      sm_xsta_addq_sel
```

6. リモート接続の SMIT セレクター値をリストするには、以下のように入力します。

```
piolsvp -Axst -nadd_queue
```

このコマンドの出力は以下のようになります。

```
sm_xsta_addq_sel
```

## ファイル

```
/usr/lib/lpd/pio/etc/piolsvp  
/etc/qconfig  
/var/spool/lpd/pio/@local/custom/*  
  
/usr/lib/lpd/pio/etc/*.attach
```

**piolsvp** コマンドが入っています。  
構成ファイルが入っています。  
カスタマイズされた仮想プリンターの属性ファイルが入っています。  
接続タイプのファイルが入っています。

## 関連情報

**piobe** コマンド、 **qprt** コマンド。

プリンターおよび印刷 の プリンター接続ファイル のセクション。

プリンターおよび印刷 の 印刷の管理 のセクション。

プリンターおよび印刷 の 印刷スプーラー のセクション。

プリンターおよび印刷 の プリンター・バックエンドのプログラム のセクション。

---

## piomgpdev コマンド

### 目的

プリンターの疑似デバイスを管理します。

### 構文

```
piomgpdev -p PseudoDevice -t AttachmentType { -A | -C | -R | -D } [ -a Clause ... ]
```

### 説明

**piomgpdev** コマンドは、プリンター接続用の疑似デバイスの変更および除去を行います。 **piomgpdev** コマンドは、疑似デバイスについての情報を **/var/spool/lpd/pio/@local/dev** ディレクトリー内のファイルに格納します。このファイルには、以下のフォーマットのスタンザが入っています。

```
key_word = value
```

これらのファイルに格納された情報は、指定の接続とプリンターの接続特性に関するものです。

## フラグ

<b>-a</b> <i>Clause</i>	疑似デバイスのファイルで追加または変更する文節を指定します。文節の形式は以下のとおりです。  key_word = value
<b>-A</b>	<b>-D</b> フラグを指定した場合、文節にはキーワードしか入れることができません。
<b>-C</b>	疑似デバイスを追加します。
<b>-D</b>	疑似デバイスを変更します。
<b>-p</b> <i>PseudoDevice</i>	疑似デバイス定義のうち指定した文節に関する情報を表示します。
<b>-R</b>	プリンター接続用の疑似デバイス名を指定します。 疑似デバイスを除去します。

## ファイル

<code>/usr/lib/lpd/pio/etc/piomgpdev</code>	<b>piomgpdev</b> コマンドが入っています。
<code>/var/spool/lpd/pio/@local/dev/*</code>	プリンターの疑似デバイスのファイルが入っています。

## 関連情報

**piobe** コマンド、 **qprt** コマンド。

プリンターおよび印刷 の 印刷の管理 のセクション。

プリンターおよび印刷 の 印刷スプーラー のセクション。

プリンターおよび印刷 の プリンター・バックエンドのプログラム のセクション。

---

## piomkapqd コマンド

### 目的

SMIT ダイアログを構築して、印刷キューとプリンターを作成します。

### 構文

#### 既存のプリンターへの印刷キューを作成する

```
piomkapqd -A AttachmentType -p Printer -d DeviceName -h Header [ -e ]
```

#### プリンターと印刷キューを作成する

```
piomkapqd -A AttachmentType -p Printer -v Device -s Subclass -r Adapter -h Header [ -e ]
```

#### TTY に接続するプリンターを作成するか、ファイルにプリンター出力を割り当てて新しいキューを作成する

```
piomkapqd -A AttachmentType -p Printer { -T TTYName | -f FileName } -h Header [ -e ]
```

#### 新しいプリンターと印刷キューにユーザー定義の接続を使用する

```
piomkapqd -A AttachmentType -p Printer [ -d DeviceName ] -c CmdExec -i DiscCmd -o ObjectID -h Header [ -e ]
```

## 説明

**piomkapqd** コマンドは、ユーザーが新しいプリンターと印刷キューを作成できるように、System Management Interface Tool (SMIT) ダイアログを作成します。 **piomkapqd** コマンドを指定すると、ユーザーがユーザー定義の接続タイプを SMIT プリンターまたはキュー定義ダイアログに追加できます。

## フラグ

<b>-A</b> <i>AttachmentType</i>	プリンターをデータ・ソースに接続するために使う接続タイプを指定します。 <i>AttachmentType</i> 変数に共通の値は、以下のとおりです。 <b>local</b> ローカル接続タイプを指定します。 <b>ascii</b> ASCII 接続タイプを指定します。 <b>file</b> データを格納するファイルを指定します。
<b>-c</b> <i>CmdExec</i>	SMIT コマンド <b>cmd_to_execute</b> の値を指定します。このフラグは、ユーザー定義の接続ダイアログを作成するときに使います。このフラグを指定しなければ、デフォルトとして <b>piomkqpq</b> コマンドが使用されます。
<b>-d</b> <i>DeviceName</i>	lp0 または tty1 など、出力先となるデバイス、疑似デバイス、またはファイルの名前を指定します。
<b>-e</b>	既存の印刷キューをプリンターの出力に使用するよう指定します。 <b>-e</b> を指定すると、 <b>piomkapqd</b> コマンドを指定しても新しいキューを作成しません。
<b>-f</b> <i>FileName</i>	出力を格納するファイル名を示します。
<b>-h</b> <i>Header</i>	作成する SMIT ダイアログの表題またはヘッダーを指定します。
<b>-i</b> <i>DiscCmd</i>	SMIT コマンド <b>cmd_to_discover</b> の値を指定します。このフラグは、ユーザー定義の接続ダイアログを作成するときに使います。このフラグを指定しない場合、 <b>piomkapqd</b> コマンドのデフォルト値を使って、ダイアログを作成します。
<b>-o</b> <i>ObjectID</i>	<i>ObjectID</i> 変数の値と一致する ID を持つ SMIT オブジェクトを指定します。
<b>-p</b> <i>Printer</i>	ibm4019 などのように、 <b>/usr/lib/lpd/pio/predef</b> ディレクトリーで定義されたプリンターのタイプを指定します。
<b>-r</b> <i>ParentAdapter</i>	プリンターの親アダプターを指定します。
<b>-s</b> <i>Subclass</i>	プリンターが属するサブクラスのタイプを指定します。 <i>Subclass</i> 変数に使用できる値は、以下のとおりです。 <ul style="list-style-type: none"><li>• <b>parallel</b></li><li>• <b>rs232</b></li><li>• <b>rs422</b></li></ul>
<b>-T</b> <i>TTYName</i>	新しいプリンターまたはキューに接続する TTY の名前を指定します。
<b>-v</b> <i>Device</i>	ODM データベース内で定義されているようにデバイス・タイプを指定します。 <b>-v</b> フラグを指定すると、 <b>/usr/lib/lpd/pio/predef</b> ディレクトリーに格納されていないプリンター定義を取り出すことができます。

## 例

- 印刷キューを既存のローカル・プリンターに追加する SMIT ダイアログを作成するには、以下のように入力します。

```
piomkapqd -A local -p ibm4019 -d lp0 -h 'Add a New Queue'
```
- 新しいプリンター lp2 とローカル接続の新しい印刷キューを追加する SMIT ダイアログを作成するには、以下のように入力します。

```
piomkapqd -A local -p ibm4019 -v ibm4019 -s rs232 -r sa0 -h 'Add New Printer'
```
- TTY に接続するプリンターを追加し、そのプリンター用のキューを新しく作成する SMIT ダイアログを作成するには、以下のように入力します。

```
piomkapqd -A tty -p ibm4039 -T tty12 -h 'Add TTY Printer'
```

- 出力をファイル名 `stuff` に送り、新しいキューを作成する `SMIT` ダイアログを作成するには、以下のように入力します。

```
piomkapqd -A file -p ibm4039 -f stuff -h 'Add Output File' -e
```

- ユーザー定義のプリンター接続タイプを追加し、新しいキューを作成する `SMIT` ダイアログを作成するには、以下のように入力します。

```
piomkapqd -A hpJetDirect -p hplj-4 [-d lp0] -c /usr/sbin/mkjetd -i /usr/bin/ljsd -o JetDirect -h 'Add New Attachment Type'
```

## ファイル

`/usr/lib/lpd/pio/etc/piomkapqd`

`piomkapqd` コマンドが入っています。

## 関連情報

`piobe` コマンド、`piomkqp` コマンド。

プリンターおよび印刷 の 印刷の管理、印刷スプーラー のセクション。

オペレーティング・システムおよびデバイスのマネージの System management interface tool のセクション。

プリンターおよび印刷 の プリンター・バックエンドのプログラム のセクション。

---

## piomkqp コマンド

### 目的

印刷キューを作成します。

### 構文

#### 新しいプリンターを追加する

```
piomkqp -A AttachmentType -p PrinterType -Q QueueName -D DataStream -v DeviceType -s Subclass  
-r ParentAdapter -w PortNumber [  
-a { interface | ptop | autoconfig | speed | parity | bpc | stops | xon | dtr | tbc=DescValue } ] ...
```

#### 新しい印刷キューを作成する

```
piomkqp -A AttachmentType -p PrinterType { -D DataStream | -q QueueName } -s Subclass  
-r ParentAdapter -w PortNumber -v DeviceType [  
-a { interface | ptop | autoconfig | speed | parity | bpc | stops | xon | dtr | tbc=DescValue } ] ...
```

#### 既存のプリンターに印刷キューを作成する

```
piomkqp -A AttachmentType -p PrinterType -d DeviceName { -D DataStream | -q QueueName }
```

#### 既存の印刷キューに既存のプリンターを追加する

```
piomkqp -A AttachmentType -p PrinterType -d DeviceName -D DataStream -q QueueName
```



## 説明

**piomkpg** コマンドは、印刷キューとプリンターを作成します。このコマンドは、**piomkapqd** コマンドを使用して作成した **SMIT** ダイアログで使用します。**piomkpg** コマンドは以下の機能を実行します。

- 各種の接続タイプを持つプリンターを作成する。
- 印刷キューを作成する。
- キュー・デバイスを作成する。
- 仮想プリンターを作成する。
- 疑似デバイスを作成する。

## フラグ

- a** デバイスの属性を指定します。このフラグは、**-a speed=9600** などのように **Attribute=Value** のフォーマットをとります。有効な属性は次のとおりです。
- Interface**
- ptop**
  - autoconfic**
  - speed**
  - parity**
  - bpc**
  - stops**
  - xon**
  - dtr**
  - tbc**
- A AttachmentType** プリンターをデータ・ソースに接続するために使う接続タイプを指定します。**AttachmentType** 変数に共通の値は、以下のとおりです。
- local** ローカル接続タイプを指定します。
  - ascii** ASCII 接続タイプを指定します。
  - file** データを格納するファイルを指定します。
- d DeviceName** lp0 または tty1 など、出力先となるデバイス、疑似デバイス、またはファイルの名前を指定します。
- D DataStream** 作成する印刷キュー、または既存の印刷キューのデータ・ストリームを指定します。
- p PrinterType** ibm4019 などのように、**/usr/lib/lpd/pio/predef** ディレクトリーで定義されたプリンターのタイプを指定します。
- q QueueName** 新しいキューの名前を指定します。**-q** および **-Q** フラグは排他的です。
- Q QueueName** 既存のキューの名前を指定します。**-q** および **-Q** フラグは排他的です。
- s Subclass** プリンターが属するサブクラスのタイプを指定します。**Subclass** 変数に使用できる値は、以下のとおりです。
- **parallel**
  - **rs232**
  - **rs422**
- r ParentAdapter** プリンターの親アダプターを指定します。
- w PortNumber** プリンター接続用のポート番号を指定します。
- v DeviceType** ODM データベース内で定義されているようにデバイス・タイプを指定します。

## 例

1. 既存の IBMIBM 4019 プリンター lp0 に使用するデータ・ストリーム ASCII のローカル印刷キュー castor を作成するには、次のように入力します。

```
piomkppq -A local -p ibm4019 -d lp0 -D asc -q castor
```

2. 既存のローカル・プリンターを、データ・ストリーム PostScript 用の既存のローカル印刷キュー pyrite に追加するには、以下のように入力します。

```
piomkppq -A local -p ibm4019 -d lp0 -Q pyrite -D ps
```

3. 新しいプリンター向けにローカル印刷キュー baker を作成するには、以下のように入力します。

```
piomkppq -A local -p ibm4019 -D asc -Q baker -s parallel -r ppa0  
-w p -v ibm4019 [-a ptop=120]
```

4. **clues** ファイルの印刷キューを作成するには、以下のように入力します。

```
piomkppq -A file -p ibm4019 -d clues -D asc -q baker
```

## ファイル

<code>/usr/lib/lpd/pio/etc/piomkppq</code>	<b>piomkppq</b> コマンドが入っています。
<code>/usr/lib/lpd/pio/etc/piomgpdev</code>	疑似デバイスを作成します。
<code>/usr/sbin/mkdev</code>	デバイスを作成します。
<code>/usr/bin/mkque</code>	キューを作成します。
<code>/usr/bin/mkquedv</code>	キュー・デバイスを作成します。
<code>/usr/sbin/mkvirprt</code>	仮想プリンターを作成します。

## 関連情報

**piobe** コマンド、**piomkapqd** コマンド。

プリンターおよび印刷 の 印刷の管理 のセクション。

プリンターおよび印刷 の 印刷スプーラー のセクション。

プリンターおよび印刷 の プリンター・バックエンドのプログラム のセクション。

---

## piomsg コマンド

### 目的

プリンター・バックエンド・メッセージをユーザーに送信します。

### 構文

```
piomsg [ -u UserList ] [ -c MsgCatalog [ -s MsgSet ] -n MsgNumber ] [ -a MsgArg ] ...  
[ MsgText ]
```

### 説明

**piomsg** コマンドは、メッセージ・カタログからプリンター・バックエンド・メッセージを取り出すか、指定したメッセージ・テキストを 1 人または複数のユーザーに送信するかのいずれかを行います。

**piomsg** コマンドは、印刷ジョブを実行すると動作します。通常 **piomsg** コマンドは、プリンターのコン・ファイルで使用して、**piobe** コマンドを指定して印刷ジョブを処理している間に、印刷ジョブを実行依頼しているユーザーにメッセージを送信します。

**-c**、**-s**、または **-n** フラグを指定すると、**piomsg** コマンドはメッセージ・カタログからメッセージを取り出します。このコマンドは、**NLSPATH** 環境変数で指定したディレクトリーでメッセージを検索します。**NLSPATH** 環境変数にディレクトリー・パスが入っていない場合、**piomsg** コマンドは、デフォルトのディレクトリー **/usr/lib/lpd/pio/etc** を検索します。**/usr/lib/lpd/pio/etc** ディレクトリー内にメッセージがない場合、そのコマンドは、*MessageText* 変数で指定したテキストを提供します。**-c**、**-s**、または **-n** フラグを指定しなければ、**piomsg** コマンドは、*MessageText* 変数の値 (ある場合) を戻します。

各メッセージは、**%s** または **%n\$s printf** サブルーチン変換の仕様について解析されます。メッセージ文字列を指定すると、メッセージがユーザーに送信される前に **printf** 変換の仕様が提供されたメッセージ文字列に置き換えられます。**piomsg** コマンドは、改行 **/n** または水平タブ **/t** など、メッセージに組み込まれているエスケープ・シーケンスを処理します。

## フラグ

<b>-a</b> <i>MsgArg</i>	メッセージ引数文字列を指定します。 <i>MsgArg</i> 変数の値は、 <b>%s</b> または <b>%n\$s printf</b> サブルーチンの変換の仕様が入っていると、メッセージに置換します。 <b>-a</b> フラグを最高 10 個まで指定して、複数の引数を指定できます。変換の仕様の解析中にエラーが発生すると、元のメッセージが送信されます。
<b>-c</b> <i>MsgCatalog</i>	取り出すメッセージが入っているメッセージ・カタログを指定します。 <b>-c</b> フラグは、 <b>-n</b> フラグとともに指定しなければなりません。
<b>-n</b> <i>MsgNumber</i>	メッセージ番号を指定します。 <b>-n</b> フラグは、 <b>-c</b> フラグとともに指定しなければなりません。
<b>-s</b> <i>MsgSet</i>	オプションのメッセージ・セットを指定します。 <i>MsgSet</i> 変数のデフォルト値は 1 です。 <b>-s</b> フラグは、 <b>-c</b> と <b>-n</b> フラグの両方とともに指定しなければなりません。
<b>-u</b> <i>UserList</i>	メッセージを受け取るユーザーのリストを指定します。 <i>UserList</i> 変数のユーザー名とノード名は、コンマで区切ります。ユーザー・リストにノード名を入れるには、@ 文字に続けてノード名またはアドレスを指定します。 <b>-u</b> フラグを省略した場合、メッセージは印刷ジョブを開始したユーザーに戻ります。

## 例

1. メッセージ・カタログ **piobe.cat** からメッセージ・セット番号 1 内のメッセージ番号 100 を取り出して、そのメッセージを印刷サーバーと同じノード上のユーザー **joe** とノード **foobar** 上のユーザー **tom** に送信するには、以下のように入力します。

```
piomsg -u joe,tom@foobar -c piobe.cat -n 100
```

2. メッセージ引数文字列が付いているメッセージを、印刷ジョブを実行依頼しているユーザーに送信するには、以下のように入力します。

```
piomsg -a "/usr/bin/troff" "The specified filter %s is not found\n"
```

3. **xyz.cat** からセット番号 2 にあるメッセージ番号 5 を取り出し、障害が発生した場合には、ダミー・メッセージを使用して、プリンターにメッセージを送信するには、以下のように入力します。

```
piomsg -cxyz.cat -s2 -n5 "xyz.cat is not installed.\n"
```

**注:** **piomsg** コマンドが **NLSPATH** 環境変数またはデフォルトのディレクトリーとともに指定されたカタログからメッセージを取り出すことができないときは、ユーザーに提供されたメッセージ・テキストを送信します。

## ファイル

`/usr/lib/lpd/pio/etc/piomsg`

`piomsg` コマンドが入っています。

## 関連情報

`piobe` コマンド。

`printf` サブルーチン。

プリンターおよび印刷 の 印刷の管理 のセクション。

プリンターおよび印刷 の 印刷スプーラー のセクション。

プリンターおよび印刷 の プリンター・バックエンドのプログラム のセクション。

---

## pioout コマンド

### 目的

プリンター・バックエンドのデバイス・ドライバ・インターフェース・プログラムです。

### 構文

```
/usr/lpd/pio/etc/pioout [ -A BytesPrinted ] [ -B TotalBytes ] [ -C NumberCancelStrings ]  
[ -D CancelString ] [ -E Mask ] [ -F FormFeedString ] [ -I InterventionRequiredUser ]  
[ -K TextString ] [ -L TextString ] [ -N NumberFormFeedStrings ] [ -O OutFile ] [ -P PrefixFile ]  
[ -R ParseRoutine ] [ -S SuffixFile ] [ -W+ ]
```

### 説明

`pioout` コマンドは、パイプラインの終了時にプリンターにファイルやバースト・ページを印刷するために `piobe` コマンド (印刷ジョブ・マネージャー) によって呼び出されます。このコマンドは、標準入力からの入力データ、接頭部ファイル (`-P` フラグが指定されている場合)、および接尾部ファイル (`-S` フラグが指定されている場合) を読み取り、次にプリンター (`-O` フラグが指定されている場合は、`OutFile`) にデータを書き込みます。介入が必要な (`-I` フラグが指定されていない場合の) エラー条件およびエラー状態は、印刷ジョブを実行依頼したユーザーに報告されます。

`-A` フラグと `-B` フラグで指定された値は、印刷ジョブがどれだけ完了したかのパーセンテージを定期的に `qdaemon` プロセスに報告するため使用されます。 `-C` フラグおよび `-D` フラグは、印刷ジョブが取り消された場合、プリンターに送信されるデータ文字列を指定します。

`-O` フラグは、ヘッダー・ページを生成して、それを一時ファイルに格納するために使用します。 `-P` フラグは、印刷ファイルを印刷する直前に (一時ファイルに保存された) ヘッダー・ページを印刷するために使用します。

`pioout` コマンドは、以下の環境変数を初期化する必要があります。

<code>PIOTITLE</code>	印刷ジョブの表題。
<code>PIODEVNAME</code>	デバイス名。
<code>PIOQNAME</code>	印刷キューの名前。
<code>PIOQDNAME</code>	キュー・デバイス名。

**PIOFROM**  
**PIOMAILONLY**  
**PIOTERM**

印刷ジョブを実行依頼したユーザー。  
ゼロ以外の場合、ユーザーあてのメッセージは常にメールされ、表示されません。  
tty 定義から想定された端末タイプを無効にします。この変数は、端末接続の端末へ実行依頼される印刷ジョブについてのみ使用されます。

## フラグ

- A BytesPrinted** 印刷ジョブについて既に印刷されたバイト数を指定します。
- B TotalBytes** 印刷ジョブについて印刷されるバイト数の合計を指定します。
- C NumberCancelStrings** 印刷ジョブが取り消されたとき、**-D** フラグで指定された文字列がプリンターに送信される回数を指定します。このフラグを指定しなければ、値は 3168 であると想定されます。
- D CancelString** 印刷ジョブが取り消されたときに、プリンターに送信される文字列を指定します。**-D** フラグが指定されていない場合、文字列は、1 つの null 文字からなるものと想定されます。
- E Mask** *Mask* として、1 つまたは複数のデバイス・ドライバのエラー・フラグ名をコマンドで区切って指定します。マスクが **LPQUERY** コマンドを持つ **ioctl** サブルーチンによって戻りものである場合、マスクによって示されたエラー条件は無視されます。フラグ名には、**LPST\_ERROR**、**LPST\_NOSLCT**、および **LPST\_SOFT** を指定でき、**/usr/include/sys/lpio.h** ファイルに定義されます。
- F FormFeed String** 用紙送りするためにプリンターに送信する文字列を指定します。**-F** フラグが指定されていない場合、文字列は `\014` であると想定されます。
- I InterventionRequiredUser** プリンターが介入を必要とするときにメッセージの送信先となるユーザーを指定します。このフラグを指定しなければ、メッセージは印刷ジョブを実行依頼したユーザーに送られます。  
  
*InterventionRequiredUser* パラメーターには、1 つまたは複数のユーザー名をコマンドで区切って使用できます。null 文字列は印刷ジョブの実行依頼側を表します。例えば、文字列 `jim@server02` を使用すると、介入が必要だというメッセージが印刷ジョブの実行依頼側とノード `server02` 上のユーザー `jim` の両方に送信されます。
- K TextString** PostScript プリンターによって送信されるメッセージが指定されたテキスト文字列に入っている場合、そのメッセージを廃棄することを指定します。例えば、*TextString* 変数が `warming up` の場合に、テキスト `warming up` を含むメッセージが廃棄されます。
- L TextString** PostScript プリンターから受信したメッセージに指定されたテキスト文字列が含まれる場合、メッセージ中のこの文字列に続くテキストが、**-I** フラグで指定された介入が必要なユーザーへ送信されます。
- N NumberFormFeedStrings** 入力データ・ストリームの終わりでプリンターに送信する用紙送り文字列数を指定します。このフラグを指定しなければ、値はゼロであると想定されます。通常このフラグは、プリンターがアイドル状態になった後で連続用紙を位置合わせするため、または、プリンターがアイドル状態になるときに用紙を送るために使います。
- O OutFile** 出力をプリンターに送信する代わりに、指定されたファイルに送信することを指定します。
- P PrefixFile** 印刷ファイルの最初のバイトが送信される前にプリンターに送信されるファイルを指定します。印刷ファイルの先頭バイトが到着する前に印刷ジョブが終了すると、接頭部ファイルは送信されません。
- R ParseRoutine** ルーチンの絶対パス名を指定して、プリンターから読み取られたデータを解析します。**/usr/include/piestruct.h** ファイルに解析ルーチンの例を入れます。**-R** フラグが指定されていない場合、デフォルトの解析ルーチンを使用します。

**-S** *SuffixFile* 印刷ファイルが送信されてから、プリンターに送られるファイルを指定します。印刷ファイルの最初のバイトが到着する前に印刷ジョブが終了すると、接尾部ファイルは送信されません。

**-W +** 終了するために EOF (16 進の 04) をプリンターから受信しなければならないことを指定します。

## 関連情報

**piobe** コマンド、**pioburst** コマンド、**piodigest** コマンド、**pioformat** コマンド、**pioquote** コマンド、**piopredef** コマンド、**qdaemon** コマンド。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の プリンター・バックエンドのプログラム のセクション。

「AIX 5L Version 5.3 Kernel Extensions and Device Support Programming Concepts」の Printer Addition Management Subsystem: Programming Overview のセクション。

プリンターおよび印刷 の プリンター・コード・ページ変換テーブル のセクション。

プリンターおよび印刷 の プリンター・コロンのファイルの規則 のセクション。

プリンターおよび印刷 の プリンター・コロンのファイルのエスケープ・シーケンス のセクション。

プリンターおよび印刷 の バックエンドと qdaemon の相互作用 のセクション。

プリンターおよび印刷 の libqb 内のバックエンド・ルーチン のセクション。

プリンターおよび印刷 の プリンター・コロンのファイルを使用したプリンターの追加 のセクション。

---

## piopredef コマンド

### 目的

事前に定義したプリンター・データ・ストリーム定義を作成します。

### 構文

```
piopredef [ -r ] -d QueueDeviceName -q PrintQueueName -s DataStreamType -t PrinterType
```

### 説明

**piopredef** コマンドは、仮想プリンターの定義から、事前定義プリンターのデータ・ストリームの定義を作成します。このコマンドは、**mkvirprt** コマンドの逆のものと考えてことができます。**mkvirprt** コマンドは、事前定義プリンターのデータ・ストリームの定義をコピーして、必要に応じてカスタマイズできる仮想プリンターの定義を作成します。これに対して、**piopredef** コマンドは、カスタマイズされた仮想プリンターの定義をコピーして、事前定義プリンターのデータ・ストリームの定義を作成します。

**piopredef** コマンドを使用すると、サポートされているプリンターの印刷データ・ストリームに類似した印刷データ・ストリームを受け入れる、サポートされていないプリンターのために事前定義プリンターの定義を作成できます。例えば、4201-3 Proprinter III に対する仮想プリンターの定義は、**mkvirprt** コマンドで



作成でき、**lsvirprt** コマンドで表示でき、必要に応じて **chvirprt** コマンドでサポートされていないプリンターを修正でき、次に **piopredef** コマンドを指定して、サポートされていないプリンターを事前に定義することができます。

**mkvirprt** コマンドを使って新しく事前定義プリンターの定義を指定して、同じコンピューター上にサポートされないプリンターのタイプに仮想プリンターを追加する、または別のコンピューターにトランスポートして使用できます。

## フラグ

<b>-d</b> <i>QueueDeviceName</i>	事前定義プリンターの定義を作成するために使用する、カスタマイズされた仮想プリンターの定義のスプーラーを <i>QueueDeviceName</i> 変数で指定します。
<b>-q</b> <i>PrintQueueName</i>	事前定義プリンターの定義を作成するために使用する、仮想プリンターの定義のスプーラーを <i>PrintQueueName</i> 変数で指定します。
<b>-r</b>	<b>-s</b> フラグと <b>-t</b> フラグを指定して、既存の事前定義プリンターの定義を指定する場合、既存のプリンター定義を置換するように指定します。
<b>-s</b> <i>DataStreamType</i>	作成する事前定義プリンター定義のプリンターを <i>DataStreamType</i> 変数で指定します。データ・ストリームのタイプの例は、以下のとおりです。
	<b>asc</b> IBM 拡張 ASCII
	<b>gl</b> Hewlett-Packard GL
	<b>pcl</b> Hewlett-Packard PCL
	<b>ps</b> PostScript
	<b>630</b> Diablo 630
	<b>855</b> Texas Instruments 855
<b>-t</b> <i>PrinterType</i>	作成する事前定義プリンター定義のプリンター・タイプを指定します。既存のプリンター・タイプには 4201-3、hplj-2、ti2115 などがあります。

注: フラグを指定しない場合、コマンド構文が表示されます。

## 例

仮想プリンターに対して既存の仮想プリンターの定義から新しく事前定義プリンターの定義を作成するには、以下のように入力します。

```
piopredef -d mypro -q proq -s asc -t 9234-2
```

proq 印刷キュー上の mypro キュー・デバイスを割り当てられた仮想プリンターの属性をコピーして 9234-2 プリンター (asc データ・ストリーム) の新しい事前定義プリンターの定義を作成します。

## ファイル

**/etc/piopredef**  
**/usr/lpd/pio/predef/\***

**/var/spool/lpd/pio/@local/custom/\***

**piopredef** コマンドが入っています。  
事前定義プリンターのデータ・ストリームの属性ファイル。ファイル名は、PrinterType.DataStreamType のフォーマットになります。  
カスタマイズされた仮想プリンターの属性ファイル。ファイル名は、PrintQueueName:QueueDeviceName のフォーマットになります。



## 関連情報

プリンターおよび印刷 の 印刷の管理、 プリンター固有の情報、 仮想プリンターの定義と属性、 プリンター・バックエンドのプログラム、 プリンター・コロン・ファイルを使用したプリンターの追加、 プリンター・コード・ページ変換テーブル のセクション。

「AIX 5L Version 5.3 Kernel Extensions and Device Support Programming Concepts」の Printer Addition Management Subsystem: Programming Overview のセクション。

---

## pkgadd コマンド

### 目的

ソフトウェア・パッケージまたはセットをシステムに転送します。

### 構文

#### ソフトウェア・パッケージをインストールする

```
pkgadd [ -d Device ] [ -r Response ] [ -n ] [ -a Admin ] [ -P Path ] [ Pkginst1 [ Pkginst2 [ . . . ] ] ]
```

#### ソフトウェア・パッケージを指定されたスプール・ディレクトリーにコピーする

```
pkgadd -s Spool [ -d Device ] [ Pkginst1 [ Pkginst2 [ . . . ] ] ]
```

### 説明

**pkgadd** は、ソフトウェア・パッケージまたはセットの内容を、システムにインストールするために配布メディアまたはディレクトリーから転送します。パッケージは、個別にインストールできる、関連するファイルおよび実行可能ファイルの集合です。セットはセット・インストール・パッケージ (SIP) と呼ばれる特殊目的のパッケージと、そのセットのメンバーである 1 つまたは複数のパッケージのコレクションからなります。SIP はセットのインストールを制御します。

**pkgadd** は、コマンド行にリストされたすべてのパッケージがインストール・メディアに含まれているかどうかを検査します。リストされたパッケージのいずれかが存在しない場合には、システムは変更されません。つまり、リストされたどのパッケージもインストールされません。

**注:** **pkgadd** コマンドを正常に実行するためには、root 以外のユーザーは以下の条件を満たしていなければなりません。

1. ユーザーは、**pkgmap** ファイル内に指定されたパスへの書き込み許可を持っていないとしない。
2. 現在の *user:group* が、**pkgmap** ファイル内で指定された *user:group* と一致しなければならない。
3. ユーザーは、**/var/sadm/install** および **/var/sadm/pkg** ディレクトリーへの書き込み許可を持っていないとしない。

**-d** フラグを指定しないで使用した場合、**pkgadd** はデフォルト・スプール・ディレクトリーからパッケージを探します (**/var/spool/pkg**)。 **-s** フラグを指定して使用した場合には、パッケージをインストールする代わりにスプール・ディレクトリーに書き込みます。

エラー・メッセージはすべてログに記録されます。さらに、**pkgadd** は、終了するときに (デフォルトでは "root" にあてて) メールを送信します。このメールには、すべてのエラー・メッセージが示され、また完全にインストールされたパッケージ、部分的にインストールされたパッケージ、またはまったくインストールされなかったパッケージが要約されます。

## フラグ

- d Device** *Device* からパッケージ/セットをインストールまたはコピーします。 *Device* には、ディレクトリー、ファイル、または名前付きパイプへの絶対パス名を指定することも、 "-" (これは標準入力から読み取られるデータ・ストリーム形式のパッケージを示します) を指定することもできます。デフォルト・デバイスはインストール先のスプール・ディレクトリー (**/var/spool/pkg**) です。
- r Response** 前に対話モードで行われた **pkgask** セッション (**pkgask** コマンドを参照) 中に「要求スクリプト」によって出された質問に対する応答を含む、ファイルまたはディレクトリー *Response* を識別します。 *Pkginst* がパッケージである場合、 *Response* には絶対パス名またはディレクトリーを指定することができ、 *Pkginst* が SIP である場合、 *Response* にはディレクトリーを指定しなければなりません。
- n** インストールが非対話式モードで行われることを指定します。デフォルト・モードは対話式です。
- a Admin** インストール時の検査 (スペース・サイズ、システム状態などの検査) を行うかどうかを指定するために、デフォルトの管理ファイルの代わりに使用される、インストール先の管理ファイル *Admin* を定義します。 "none" というトークンを指定すると、 **admin** ファイルの使用がオーバーライドされ、ユーザーとの対話が強制されます。絶対パス名を指定しない限り、 **pkgadd** は **/var/sadm/install/admin** ディレクトリーからファイルを探します。デフォルトでは、そのディレクトリー内のファイル **default** が使用されます。 **default** は、パッケージをインストールするための十分なスペースがあるかどうか、または他のパッケージへの依存性があるかどうかを調べる場合を除き、検査が行われないことを指定します。 *Pkginst* が SIP である場合には、 **-a** フラグは使用できません。
- P Path** インストールのための代替ルート・ディレクトリー・パスを指定します。ファイルはこの場所の下にインストールされます。
- Pkginst* パッケージ/セット名の省略形を指定するために使用する短文字列を定義します。(「パッケージ・インスタンス」という用語は、厳密な意味で使用されているのではなく、 *Pkginst* のすべてのインスタンス化を指します。) **pkginfo** コマンドおよび **pkginfo** ファイル・フォーマットを参照してください。
- Pkginst* が SIP である場合、SIP 制御は、要求スクリプトおよびプリインストール・スクリプトを使用して、セットのインストールを制御します。ユーザーに応答の入力をプロンプト指示したり適切なアクションを行ったりするのは、パッケージ・インストール・ツールではなく、SIP 要求スクリプトです。要求スクリプトが失敗した場合、SIP だけが処理されます。
- パッケージのすべてのインスタンスを指示するには、シェルが "\*" 文字を解釈しないようにするために、単一引用符でコマンド行を囲んで '*Pkginst.\**' と指定してください。ソース・メディアで使用可能なすべてのパッケージを参照するには、"all" というトークンを使用してください。
- s Spool** パッケージをインストールする代わりに、ディレクトリー *Spool* に読み込みます。

## 特別な注

**-r** フラグを使用すると、ファイル名と一緒にディレクトリー名を指示できます。ディレクトリーに非常に多くの *Response* ファイルを含めて、各ファイルで、関連付けられるパッケージの名前を共用できます。例えば、**pkgadd** を 1 回起動するだけで複数の対話式パッケージを追加する場合に、この方法を使用できま

す。要求スクリプトを含む各パッケージには、*Response* ファイルが必要です。パッケージと同じ名前の応答ファイル (例えば、*Package1* や *Package2*) を作成した場合、**-r** フラグの後に、それらのファイルが入っているディレクトリーの名前を指定してください。

**-n** フラグを使用すると、インストールを完了させるためになんらかの対話が必要な場合には、インストールは一時停止します。

コマンド行に *Pkginst* を指定しないで **pkgadd** を起動すると、メディアに少なくとも 1 つ SIP が存在している場合には、セットの名前だけが表示されます。したがって、セットのメンバーであるパッケージとそうでないパッケージとを、同じメディアには含めないでください。このようにすると、セットのメンバーでないパッケージは、その **pkginst** 名がコマンド行で指定されていなければインストールできなくなります。

**pkgadd** コマンドは、*Pkginst* 内のファイルのうちで、既にシステムにインストールされているものがあるかどうかを検査し、既にインストールされているものがある場合には、インストールを続行する前にその事実を保管します。その後、**pkgadd** はそのようなファイルをシステムにインストールしません。いずれかのパッケージ・インストール・スクリプトがそのようなファイルを除去した場合、パッケージ・インストールが完了すると、そのファイルはシステムに存在しなくなります。

**pkgadd** コマンドは、**pkgmk** で処理される前に既に圧縮されていたファイル (つまり、**.Z** 形式のファイル) については圧縮解除を行いません。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0 スクリプトの正常終了です。
- 1 致命的なエラーです。この時点でインストール・プロセスが終了します。
- 2 警告、またはエラー条件の可能性があります。インストールは続行されます。完了時に警告メッセージが表示されます。
- 3 スクリプトは中断されました。スクリプトが未完了のままになっている可能性があります。インストールはこの時点で終了します。
- 4 スクリプトが中断されました (管理)。インストールはこの時点で終了します。
- 5 スクリプトが中断されました (対話が必要でした)。インストールはこの時点で終了します。
- 10 選択されたすべてのパッケージのインストールが完了してから、システムをリポートする必要があります。(この値は、上記の 1 桁の終了コードのいずれかに追加されます。)
- 20 システムは現行パッケージのインストールが完了したら、すぐにリポートされなければなりません。(この値は、上記の 1 桁の終了コードのいずれかに追加されます。)
- 77 セットのためのパッケージが選択されていません。
- 99 内部エラー

## ファイル

**/var/sadm/install/admin/default**  
**/var/sadm/install/logs/pkginst.log**  
**/var/spool/pkg**

デフォルトのパッケージ管理ファイル  
エラー・メッセージ・ログ  
デフォルトのスパール・ディレクトリー

## 関連情報

**pkgask** コマンド、**pkgchk** コマンド、**pkginfo** コマンド、**pkgmk** コマンド、**pkgparam** コマンド、**pkgrm** コマンド。

---

## pkgask コマンド

### 目的

要求スクリプトへの応答を保管します。

### 構文

```
pkgask [ -d Device ] -r Response [ Pkginst [ Pkginst [ . . . ] ]
```

### 説明

**pkgask** を使用すると、管理者は、対話式パッケージ (要求スクリプトを含むもの) またはパッケージのセットへの応答を保管することができます。セットはセット・インストール・パッケージ (SIP) と呼ばれる特殊目的のパッケージと、そのセットのメンバーである 1 つまたは複数のパッケージのコレクションからなります。SIP はセットのインストールを制御します。

**pkgask** を起動すると、インストール時に入力として使用される *Response* ファイルが生成されます。この *Response* ファイルを使用すると、パッケージに必要なすべての情報が既にファイルに含まれるため、インストール時に対話を行わずに済みます。

**pkgask** を実行すると、応答ファイルのほかに以下のディレクトリーが作成されます。

<b>/ptfvars</b>	パッケージに関する変数が含まれます。
<b>/fileinfo</b>	パッケージに関するチェックサム情報が含まれます。
<b>/oldfiles</b>	以前のバージョンのパッケージのバックアップが含まれます。

非対話式で他のシステムにパッケージをインストールするためには、これらのすべてのファイルおよびディレクトリーをターゲット・システムにコピーしなければなりません。

**注:** 例えば、非対話式で他のパッケージをインストールするなどの目的で、これらのディレクトリーのいずれかを上書きした場合、最初にオリジナルのディレクトリー内容を復元しなければ、最初のパッケージを正常に除去できません。

**-r** フラグを使用して、ファイル名と一緒にディレクトリー名を指示できます。ディレクトリー名を使用すると、非常に多くの *Response* ファイルを作成し、各ファイルで、関連付けられるパッケージの名前を共用できます。この方法は、例えば、**pkgadd** を 1 回起動するだけで複数の対話式パッケージを追加できるようにしたい場合などに使用できます。それぞれのパッケージに *Response* ファイルが必要です。パッケージ・インスタンスと同じ名前の複数の応答ファイルを作成するためには、**pkgask** コマンドでファイルの作成先ディレクトリーの名前を指定し、複数のインスタンス名を指定してください。パッケージをインストールするときには、このディレクトリーを **pkgadd** コマンドで指定できます。

### フラグ

**-d Device** *Device* でパッケージの要求スクリプトを実行します。*Device* には、ディレクトリーへの絶対パス名 (**/var/tmp** など) を指定することも、**"-"** (これは標準入力から読み取られるデータ・ストリーム形式のパッケージを示します) を指定することもできます。デフォルト・デバイスはインストール先のスプール・ディレクトリー (**/var/spool/pkg**) です。

**-r** *Response*

ファイルまたはディレクトリー *Response* を識別します。これは、パッケージ要求スクリプトとの対話への応答を含めるために作成する必要があります。このファイル、またはファイルのディレクトリーは、後で **pkgadd** コマンドへの入力として使用できます (**pkgadd** コマンドを参照してください)。 *Pkginst* がパッケージである場合、*Response* には絶対パス名またはディレクトリーを指定することができ、*Pkginst* が SIP である場合、*Response* にはディレクトリーを指定しなければなりません。

*Pkginst*

パッケージ/セット名を省略形で指定するために使用する短文字列を定義します。（「パッケージ・インスタンス」という用語は、厳密な意味で使用されているのではなく、インスタンス ID が指定されていないものも含め、*Pkginst* のすべてのインスタンス化を指します。）

パッケージ名の省略形を作成するには、“**PKG**” パラメーターでその省略形を指定してください。例えば、Advanced Commands パッケージに “cmds” という省略形を割り当てたい場合には、**PKG=cmds** と入力してください。

*Pkginst* で SIP を指定したときには、そのセットのメンバーであるパッケージの要求スクリプトが（存在する場合には）すべて実行され、それによって作成された応答ファイルは、**-r** フラグで指定されたディレクトリーに入ります。

パッケージのすべてのインスタンスを指示するには、シェルが “\*” 文字を解釈しないようにするために、単一引用符でコマンド行を囲んで ‘*Pkginst*.\*’ と指定してください。ソース・メディアで使用可能なすべてのパッケージを参照するには、“all” というトークンを使用してください。

**注:** コマンド行に *Pkginst* を指定しないで **pkgask** を起動すると、デバイスに少なくとも 1 つ SIP が存在している場合には、セットの名前だけが表示されます。したがって、セットのメンバーでないパッケージがある場合、それらのパッケージは、その *Pkginst* 名がコマンド行で指定されていない場合は参照できません。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0 スクリプトの正常終了です。
- 1 致命的なエラーです。この時点でインストール・プロセスが終了します。
- 2 警告、またはエラー条件の可能性があります。インストールは続行されます。完了時に警告メッセージが表示されます。
- 3 スクリプトは中断されました。スクリプトが未完了のままになっている可能性があります。インストールはこの時点で終了します。
- 4 スクリプトが中断されました（管理）。インストールはこの時点で終了します。
- 5 スクリプトが中断されました（対話が必要でした）。インストールはこの時点で終了します。
- 10 選択されたすべてのパッケージのインストールが完了してから、システムをリブートする必要があります。（この値は、上記の 1 桁の終了コードのいずれかに追加されます。）
- 20 システムは現行パッケージのインストールが完了したら、すぐにリブートされなければなりません。（この値は、上記の 1 桁の終了コードのいずれかに追加されます。）
- 77 セットのためのパッケージが選択されていません。
- 99 内部エラー

## ファイル

**/var/spool/pkg**

デフォルトのスパール・ディレクトリー



## 関連情報

**pkgadd** コマンド、**pkgchk** コマンド、**pkginfo** コマンド、**pkgmk** コマンド、**pkgparam** コマンド、**pkgtrans** コマンド、**pkgrm** コマンド。

---

## pkgchk コマンド

### 目的

インストールの正確度を検査します。

### 構文

#### インストールされたオブジェクトの内容を検査する

```
pkgchk [ -l | -a -c -f -q -v ] [ -n -x ] [ -P path ] [ -p Path1[,Path2 . . . ] ] [ -i File ] [ Pkginst . . . ]
```

#### 指定されたデバイスにスプールされたパッケージの内容を検査する

```
pkgchk -d Device [ -l | -v ] [ -p Path1[,Path2 . . . ] ] [ -i File ] [ Pkginst . . . ]
```

#### 指定された **pkgmap** に記述されたパッケージの内容を検査する

```
pkgchk -m Pkgmap [ -e Envfile ] [ -l | -a -c -f -q -v ] [ -n -x ] [ -i File ] [ -p Path1[,Path2 . . . ] ]
```

### 説明

**pkgchk** は、インストールされたファイルの正確度を検査し、あるいは **-l** フラグを使用してパッケージ・ファイルに関する情報を表示します。このコマンドはディレクトリー構造およびファイルの整合性を検査します。矛盾がある場合には、問題の詳しい説明とともに **stderr** に報告されます。

上に定義された最初の形式は、現在システムにインストールされているオブジェクトの内容または属性、あるいはその両方をリストまたは検査するために使用されます。パッケージ名をコマンド行にリストすることも、デフォルトでマシン全体の内容を検査させることもできます。パッケージが、**-P** オプションを指定した **pkgadd** コマンドを使用して代替ルート・ディレクトリー・パスにインストールされている場合は、**-P** オプションで指定された、その同じ代替ルート・ディレクトリー・パスを使用して、内容と属性を検査またはリストすることができます。

2 番目の形式は、指定されたデバイスにスプールされていて、インストールされていないパッケージの内容を、リストまたは検査するために使用されます。スプールされたパッケージについては、属性が検査できませんので、注意してください。

3 番目の形式は、指定された *Pkgmap* で記述されたオブジェクトの内容または属性、あるいはその両方をリストまたは検査するために使用されます。

### フラグ

- l** パッケージを構成する、選択されたファイルに関する情報をリストします。これは、**a**、**c**、**f**、**g**、または **v** フラグと一緒に使用できません。
- a** ファイル属性の監査だけを行い、ファイル内容は検査しません。デフォルトでは、両方を検査します。
- c** ファイル内容の監査だけを行い、ファイル属性は検査しません。デフォルトでは、両方を検査します。

- f** 可能な場合、ファイル属性を訂正します。 **-x** フラグと一緒に使用すると、隠しファイルは除去されます。このフラグを指定して **pkgchk** を起動すると、ディレクトリー、名前付きパイプ、リンク、およびスペシャル・デバイスがまだ存在しない場合には、それらが作成されます。
- q** 抑止モードを使用可能にします。存在しないファイルに関するメッセージは生成されません。
- v** 詳細モードを使用可能にします。処理されたファイルがリストされます。
- n** 揮発性のファイルまたは編集可能なファイルは無視されます。ほとんどのポストインストール検査には、このフラグを使用してください。
- x** 排他的ディレクトリーだけを検索し、インストール・ソフトウェア・データベースまたは指定された *Pkgmap* ファイルにないファイルが存在しているかどうかを調べます。(排他的ディレクトリーとは、パッケージにより、パッケージのために作成されたディレクトリーであり、パッケージとともに配送されるファイルだけを含める必要があります。パッケージ以外のファイルが排他的ディレクトリーから検索されると、**pkgchk** はエラーを報告します。) **-x** を **-f** フラグと一緒に使用すると、隠しファイルが除去されるだけで、それ以外の検査は行われません。  
 注: 隠しファイルの除去だけを行うためには、**-f** フラグと **-x** フラグと一緒に使用してください。隠しファイルを除去し、ファイルの属性と内容を検査するためには、**-f**、**-x**、**-c**、および **-a** フラグと一緒に使用してください。
- p** リストされた 1 つまたは複数のパス名の正確度だけを検査します。"pathname" には、1 つのパス名、またはコンマ (リストが引用符で囲まれている場合には、空白) で区切られた複数のパス名を指定できます。
- i** *File* からパス名のリストを読み取り、このリストを、インストール・ソフトウェア・データベースまたは指定された *Pkgmap* ファイルと対比します。"inputfile" に含まれていないパス名は検査されません。
- d** スプールされたパッケージが入っているデバイスを指定します。 *Device* には、ディレクトリーのパス名を指定することも、 "-" (これは標準入力から読み取られるデータ・ストリーム形式のパッケージを示します) を指定することもできます。
- m** パッケージを *pkgmap* ファイル *Pkgmap* と対比するように要求します。
- e** 指定された *pkgmap* ファイル内で使用されているパラメーターを解決するために、*Envfile* として指定された *pkginfo* ファイルを使用するように要求します。
- Pkginst* パッケージ名の省略形を指定するために使用する短文字列を定義します。(「パッケージ・インスタンス」という用語は、厳密な意味で使用されているのではなく、インスタンス ID が指定されていないものも含め、*Pkginst* のすべてのインスタンス化を指します。)  
  
 パッケージのすべてのインスタンスを指示するには、シェルが "\*" 文字を解釈しないようにするために、単一引用符でコマンド行を囲んで '*Pkginst.\**' と指定してください。ソース・メディアで使用可能なすべてのパッケージを参照するには、"all" というトークンを使用してください。
- P path** 代替ルート・ディレクトリー・パス内のパッケージを検査するよう要求します。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0 スクリプトの正常終了です。
- 1 致命的なエラーです。この時点でインストール・プロセスが終了します。

## ファイル

**/usr/sbin/pkgchk** **pkgchk** コマンドが入っています。

## 関連情報

**pkgadd** コマンド、**pkgask** コマンド、**pkginfo** コマンド、**pkgmk** コマンド、**pkgrm** コマンド、**pkgtrans** コマンド。



**pkginfo** ファイル・フォーマット、**pkgmap** ファイル・フォーマット。

---

## pkginfo コマンド

### 目的

ソフトウェア・パッケージまたはセット、あるいはその両方の情報を表示します。

### 構文

#### インストールされたパッケージに関する情報を表示する

```
pkginfo [ -q] [ -x | -l] [ -r] [ -p | -i] [ -a Arch] [ -P Path] [ -v Version] [ -c Category1 [,Category2 [, . . .]]] [ Pkginst [, Pkginst [, . . .]]]
```

#### 指定されたデバイスに含まれるパッケージに関する情報を表示する

```
pkginfo [ -d Device] [ -q] [ -x | -l] [ -a Arch] [ -P Path] [ -v Version] [ -c Category1 [,Category2 [, . . .]]] [ PkginstPkginst [, Pkginst [, . . .]]]
```

### 説明

**pkginfo** は、システムにインストールされたソフトウェア・パッケージまたはセット (最初の形式で要求された場合)、もしくはディレクトリーにあるソフトウェア・パッケージまたはセット (2 番目の形式で要求された場合) に関する情報を表示します。パッケージは、個別にインストールできる、関連するファイルおよび実行可能ファイルの集合です。セットはセット・インストール・パッケージ (SIP) と呼ばれる特殊目的のパッケージと、そのセットのメンバーである 1 つまたは複数のパッケージのコレクションからなります。SIP はセットのインストールを制御します。

フラグを指定しないで **pkginfo** を実行すると、インストールされたパッケージのうち、カテゴリーの値が "set" ではない各パッケージについて (それが完全にインストールされているのか、部分的にインストールされているのかを問わず)、1 行ずつ情報が表示されます。表示される情報には、パッケージの基本カテゴリー、パッケージ・インスタンス、および名前が含まれます。UNIX System V リリース 4 以前に作成された UNIX ソフトウェア・パッケージの場合、**pkginfo** はパッケージ名と省略形だけを表示します。

**-p** および **-i** フラグは、**-d** フラグと一緒に使用した場合には無意味になります。**-p** フラグと **-i** フラグは相互に排他的です。**-x** フラグと **-l** フラグは相互に排他的です。

### フラグ

<b>-q</b>	抑止モードを使用可能にします。情報は何も表示されません。このフラグは <b>-x</b> 、 <b>-l</b> 、 <b>-p</b> 、および <b>-i</b> フラグをオーバーライドします。(あるパッケージがインストールされているかどうかを照会するために、プログラムによって起動できます。)
<b>-x</b>	指定されたパッケージについて、省略形、名前、および (使用可能な場合には) アーキテクチャーとバージョンを抽出し、表示します。
<b>-l</b>	指定されたパッケージ (複数の場合もあります) について、「詳細な形式」レポート (つまり、使用可能なすべての情報が記載されたレポート) を表示します。
<b>-r</b>	指定されたパッケージが再配置可能である場合には、そのパッケージのインストール・ベースをリストします。
<b>-p</b>	部分的にインストールされたパッケージに関する情報だけを表示します。
<b>-i</b>	完全にインストールされたパッケージに関する情報だけを表示します。
<b>-a Arch</b>	パッケージのアーキテクチャーを <i>Arch</i> と指定します。

<b>-P Path</b>	代替ルート・ディレクトリー・パスにインストールされたパッケージの情報を表示します。
<b>-v Version</b>	パッケージのバージョンを <i>Version</i> と指定します。バージョン名の前にティルド "~" を付けることにより、互換性のあるすべてのバージョンを要求することができます。
<b>-c Category . . .</b>	カテゴリー <i>Category</i> に属するパッケージに関する情報を表示します。(カテゴリーは <b>pkginfo</b> ファイルのカテゴリー・フィールドで定義されます。詳細は、 <b>pkginfo</b> ファイル・フォーマットを参照してください。) コンマで区切ったリストを使用して、複数のカテゴリーを指定できます。1 つのパッケージは、複数のカテゴリーが指定されている場合でも、1 つのカテゴリーだけに属している必要があります。パッケージとカテゴリーの突き合わせでは、大文字小文字の区別は行われません。
<b>Pkginst</b>	指定されたカテゴリーが "set" である場合、 <b>pkginfo</b> はセット・インストール・パッケージ (SIP) に関する情報を表示します。 パッケージ/セット名の省略形を指定するために使用する短文字列を定義します。(「パッケージ・インスタンス」という用語は、厳密な意味で使用されているのではなく、インスタンス ID が指定されていないものも含め、 <i>Pkginst</i> のすべてのインスタンス化を指します。)  パッケージのすべてのインスタンスを指示するには、シェルが "*" 文字を解釈しないようにするために、単一引用符でコマンド行を囲んで ' <i>Pkginst.*</i> ' と指定してください。ソース・メディアで使用可能なすべてのパッケージを参照するには、"all" というトークンを使用してください。
<b>-d Device</b>	<i>Pkginst</i> が SIP である場合、その SIP と関連付けられているパッケージに関する情報が表示されます。 <i>Device</i> に入っているパッケージまたはセットから得られた情報を表示します。 <i>Device</i> には、ディレクトリーへの絶対パス名 ( <i>/var/tmp</i> など) を指定することも、"- " (これは標準入力から読み取られるデータ・ストリーム形式のパッケージを示します) を指定することもできます。デフォルト・デバイスはインストール先のスプール・ディレクトリー ( <i>/var/spool/pkg</i> ) です。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0 スクリプトの正常終了です。
- 1 致命的なエラーです。この時点でインストール・プロセスが終了します。

## ファイル

*/var/spool/pkg* デフォルトのスプール・ディレクトリー

## 関連情報

**pkgadd** コマンド、**pkgask** コマンド、**pkgchk** コマンド、**pkgmk** コマンド、**pkgrm** コマンド、**pkgtrans** コマンド。

**pkginfo** ファイル・フォーマット、**setinfo** ファイル・フォーマット。

---

## pkgmk コマンド

### 目的

インストール可能パッケージを作成します。

### 構文

```
pkgmk [ -c ] [ -o ] [ -a Arch ] [ -b BaseDir ] [ -d Directory ] [ -f Prototype ] [ -l Limit ]  
[ -p PStamp ] [ -r RootPath ] [ -v Version ] [ Variable=Value ... ] [ PkgInst ]
```

### 説明

**pkgmk** は、**pkgadd** コマンドへの入力として使用されるインストール可能パッケージを作成します。パッケージは、個別にインストールできる、関連するファイルおよび実行可能ファイルの集合です。パッケージの内容は、ディレクトリー構造形式になっています。

**pkgmk** コマンドは、パッケージ・プロトタイプ・ファイルを入力として使用し、**pkgmap** ファイルを作成します。プロトタイプ・ファイル内の各エントリーの内容は、該当の出力ロケーションにコピーされます。内容に関する情報 (チェックサム、ファイル・サイズ、変更日付) は、計算され、プロトタイプ・ファイルに指定された属性情報と一緒に **pkgmap** ファイルに保管されます。

### フラグ

**-a Arch**

*Arch* を使用して **pkginfo** ファイルに提供されたアーキテクチャー情報をオーバーライドします。

**-b BaseDir**

指定された *BaseDir* を前に付加して、ソース・マシン上に再配置可能オブジェクトを配置します。

**-c**

情報でないファイルを圧縮します。**-c** を使用する場合は **-r** オプションも指定する必要があります。 *RootPath* 指定より上の相対パスを参照する *Prototype* ファイル内のエントリーは、圧縮されません。 **pkgmk** により処理される前に既に圧縮されているファイルは (つまり、".Z" 形式のみ)、 **pkgadd** コマンドにより圧縮解除されません。

**-d Directory**

*Directory* にパッケージを作成します。指定するディレクトリーは存在するものでなければなりません。

**-f Prototype**

ファイル *Prototype* をコマンドへの入力として使用します。このファイルのデフォルト名は **Prototype** または **prototype** です。

**pkgproto** を使用して *Prototype* ファイルを作成することができます。この場合、パッケージ内で使用するすべてのインストール・スクリプトとファイルのエントリーを手作業で追加する必要があります。使用するファイルとスクリプトのエントリーだけが必要です。ただし、パッケージには必ず **pkginfo** ファイル用のエントリーを追加する必要があります。詳細は、**pkgproto** を参照してください。

**-l** *Limit*

*Limit* として、出力デバイスの最大サイズを 512 バイトのブロック単位で指定します。デフォルトでは、出力ファイルがディレクトリーまたはマウント可能なデバイスの場合、**pkgmk** は **df** コマンドを使用して、出力デバイス上で使用可能なスペースを動的に計算します。**pkgtrans** と組み合わせて、データ・ストリーム・フォーマットでパッケージを作成するのに便利です。

**-o**

同じインスタンスを上書きします。パッケージ・インスタンスが既に存在している場合には、上書きされます。

**-p** *PStamp*

**pkginfo** ファイル内の実動スタンプ定義を *PStamp* を使用してオーバーライドします。

**-r** *RootPath*

ソース・マシン上のオブジェクトを探すために指定された *RootPath* に、*Prototype* ファイル内のソース・パス名を付加します。

**-v** *Version*

*Version* を使用して、**pkginfo** ファイルに指定されたバージョン情報をオーバーライドします。

*Variable=Value*

指定された変数をパッケージ環境に入れます。

*PkgInst*

パッケージ名の省略形を指定するために使用する短文字列。**pkgmk** はバージョンまたはアーキテクチャー (またはその両方) が異なる場合、自動的に新しいインスタンスを作成します。ユーザーはパッケージの省略形のみを指定すべきであり、ユーザーが上書きする場合を除き、特定のインスタンスを指定すべきではありません。

## 例

1. **lsps** および **lsuser** コマンドを含む **mypkgA** という名前のパッケージを作成する場合、まず最初にパッケージの目次を作成する必要があります。以下に例を示します。

```
mkdir -p /home/myuser/example/pkgmk/sbin
cp /usr/sbin/lsps /home/myuser/example/pkgmk/sbin
cp /usr/sbin/lsuser /home/myuser/example/pkgmk/sbin
```

次に、**pkginfo** ファイルを作成します。この例では、**pkginfo** ファイルは `/home/myuser/example/pkgmk/pkginfo` であり、これは以下の内容を含みます。

```
PKG="mypkgA"
NAME="My Package A"
ARCH="PPC"
RELEASE="1.0"
VERSION="2"
CATEGORY="Application"
PSTAMP="AIX 2001/02/05"
```

次に、*Prototype* ファイルとして `/home/myuser/example/pkgmk/prototype` ファイルを作成し、これは以下の内容を含みます。

```
!search /home/myuser/example/pkgmk/sbin
i pkginfo=/home/myuser/example/pkgmk/pkginfo
d example /example 1777 bin bin
d example /example/pkgmk 1777 bin bin
d example /example/pkgmk/sbin 1777 bin bin
f example /example/pkgmk/sbin/lsps 555 bin bin
f example /example/pkgmk/sbin/lsuser 555 bin bin
```

次に、上の *Prototype* および **pkginfo** ファイルを使用して、**pkgmk** コマンドによりパッケージを作成します。

```
pkgmk -d /tmp -f /home/myuser/example/pkgmk/prototype
```

以下の出力が作成されます。

```
Building pkgmap from package prototype file
## Processing pkginfo file
    WARNING:parameter <CLASSES> set to "example"

## Attempting to volumize 5 entries in pkgmap
Part 1 -- 218 blocks, 10 entries
/tmp/mypkgA/pkgmap
/tmp/mypkgA/pkginfo
/tmp/mypkgA/root/example/pkgmk/sbin/lsp
/tmp/mypkgA/root/example/pkgmk/sbin/luser
## Packaging complete
```

これで、mypkgA という名前のパッケージが /tmp/mypkgA に新しく作成されました。

## 終了状況

0	スクリプトの正常終了です。
1	致命的なエラーです。この時点でインストール・プロセスが終了します。
99	内部エラー

## ファイル

/usr/sbin/pkgmk                      pkgmk コマンドが入っています。

## 関連情報

**installp** コマンド、**pkgadd** コマンド、**pkgask** コマンド、**pkgchk** コマンド、**pkginfo** コマンド、**pkgparam** コマンド、**pkgproto** コマンド、**pkgrm** コマンド、**pkgtrans** コマンド、**tar** コマンド。

**pkginfo** ファイル・フォーマット。

**installp** コマンドを使用して、インストールするアプリケーションを準備する方法については、*AIX 5L* バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグのインストール用ソフトウェアのパッケージ作成のセクションを参照してください。

---

## pkgparam コマンド

### 目的

パッケージのパラメーター値を表示します。

### 構文

**pkginfo** に含まれているパラメーターの値を表示する

```
pkgparam [ -v ] [ -d Device ] [ -P path ] Pkginst [ Param ...]
```

デバイスに含まれているパラメーターの値を表示する

```
pkgparam -d Device [ -v ] [ Param ...]
```

## ファイルに含まれているパラメーターの値を表示する

**pkgparam** -f *File* [ -v ] [ *Param* ...]

### 説明

**pkgparam** は、コマンド行で要求された 1 つまたは複数のパラメーターに関連した値を表示します。これらの値は、**pkginfo** ファイル (*Pkginst* の場合)、**-d** フラグで指定された *Device*、または **-f** フラグで指定された特定のファイルのいずれかに入っています。(2 番目の形式のように) *Device* を指定して、*Pkginst* を指定しなかった場合、*Device* にあるすべてのパッケージに関するパラメーター情報が表示されます。

パッケージが、**-P** オプションを指定した **pkgadd** コマンドを使用して代替ルート・ディレクトリー・パスにインストールされている場合は、**-P** オプションで指定された、その同じ代替ルート・ディレクトリー・パスを使用してパッケージ・パラメーターを要求することができます。

1 つのパラメーター値が 1 行ずつ表示されます。**-v** フラグが指定されていない場合には、1 つのパラメーターの値だけが表示されます。このフラグを使用すると、コマンドの出力は次の形式で示されます。

```
Parameter1='Value1'  
Parameter2='Value2'  
Parameter3='Value3'
```

コマンド行でパラメーターを指定しない場合、そのパッケージに関連したすべてのパラメーターの値が表示されます。

### フラグ

- v** 詳細モードを指定します。パラメーターの名前とその値が表示されます。
- d** *Device* *Pkginst* が保管されている *Device* を指定します。*Device* には、ディレクトリーへの絶対パス名 (*/var/tmp* など) を指定することも、*-* (これは標準入力から読み取られるデータ・ストリーム形式のパッケージを示します) を指定することもできます。
- f** コマンドに、*File* からパラメーター値を読み取ることを要求します。このファイルは **pkginfo** ファイルと同じフォーマットになっている必要があります。例えば、このようなファイルをパッケージ開発時に作成し、このステージでソフトウェアをテストするときを使用できます。
- Pkginst* パラメーター値を表示すべき特定のパッケージを定義します。あるパッケージのすべてのインスタンスを表すためには、*Pkginst.\** という形式を使用できます。この形式を使用するときには、シェルが *"\*"* 文字を解釈しないようにするために、単一引用符でコマンド行を囲んでください。
- Param* 値を表示すべき特定のパラメーターを定義します。
- P** *path* 代替ルート・ディレクトリー・パス内で **pkginfo** ファイルを検索します。

### 終了状況

指定されたパッケージに関してパラメーター情報が利用できない場合には、コマンドはゼロ以外の状況で終了します。

- 0** スクリプトの正常終了です。
- 1** 致命的なエラーです。この時点でインストール・プロセスが終了します。

### ファイル

*/var/spool/pkg* デフォルトのスパール・ディレクトリー  
*/usr/sbin/pkgparam* **pkgparam** コマンドが入っています。



## 関連情報

**pkgtrans** コマンド。

**pkginfo** ファイル・フォーマット。

---

## pkgproto コマンド

### 目的

プロトタイプ・ファイルを生成します。

### 構文

```
pkgproto [ -i ] [ -c Class ] [Path1 [=Path2 ] ...]
```

### 説明

**pkgproto** コマンドは、指示されたパスをスキャンし、**pkgmk** コマンドへの入力として使用される可能性のあるプロトタイプ・ファイルを生成します。このファイルを生成するには、このコマンドの標準出力をファイルに宛先変更する必要があります。このファイルは **pkgmk** を呼び出すときに使用できます。

コマンド行でパス が指定されていないと、標準入力パス のリストと考えられます。コマンド行にリストされたパス がディレクトリーの場合は、ディレクトリーの内容が検索されます。ただし、入力が標準入力から読み込まれる場合は、パスに指定したディレクトリーは検索されません。

プロトタイプ・ファイル属性 *mac*、*fixed*、および *inherited* は、**pkgproto** には決められないので、手作業でファイルに追加する必要があります。

デフォルトで、**pkgproto** は検出されたシンボリック・リンク (*ftype=s*) にシンボリック・リンク・エントリーを作成します。**-i** フラグを使用する場合、**pkgproto** はシンボリック・リンク (*ftype=f*) にファイル・エントリーを作成します。プロトタイプ・ファイルを編集して、*v* (揮発 (*volatile*))、*e* (編集可能 (*editable*))、または *x* (排他的ディレクトリー (*exclusive directory*)) などのファイル・タイプを割り当てる必要があります。**pkgproto** は、リンクされたファイルを検出します。複数ファイルが共にリンクされている場合は、検出された最初のパスがリンクの元と見なされます。

このコマンドからの出力は標準出力に送信されます。**pkgmk** を呼び出すときにプロトタイプ・ファイルとしてその結果を使用したい場合は、標準出力をファイルに宛先変更する必要があります。**pkgmk** は、プロトタイプ・ファイルのデフォルトのファイル名として *prototype* を使用するので、**pkgproto** の出力の宛先をファイル名 *prototype* にすることをお勧めします。

パッケージが必要とするインストール・スクリプトおよびファイル用に、このコマンドによって生成されるプロトタイプ・ファイルにエントリーを追加する必要があります。最低でも、**pkginfo** ファイルへのエントリーが必要です。パッケージ内で使用するファイル (**copyright**、**compver**、**depend**、**setinfo**、**space**、パッケージ用に定義するすべてのインストール・スクリプトまたは除去スクリプト、定義するすべてのクラス) のどれについても、エントリーが必要です。

#### 注:

- デフォルトで、**pkgproto** は検出されたシンボリック・リンク (*ftype=s*) にシンボリック・リンク・エントリーを作成します。**-i** オプションを使用すると、**pkgproto** はシンボリック・リンク用のファイル・エントリーを作成します (*ftype=f*)。プロトタイプ・ファイルを編集して、*v* (揮発 (*volatile*))、*e* (編集可能 (*editable*))、または *x* (排他的ディレクトリー (*exclusive directory*)) などのファイル・タイプを割



り当てる必要があります。**pkgproto** は、リンクされたファイルを検出します。複数ファイルが共にリンクされている場合は、検出された最初のパスがリンクの元と見なされます。

- このコマンドからの出力は標準出力に送信されます。**pkgmk** を呼び出すときにプロトタイプ・ファイルとしてその結果を使用したい場合は、標準出力をファイルに宛先変更する必要があります。**pkgmk** は、プロトタイプ・ファイルのデフォルトのファイル名として **prototype** を使用するので、**pkgproto** の出力の宛先をファイル名 **prototype** にすることをお勧めします。
- パッケージが必要とするインストール・スクリプトおよびファイル用に、このコマンドが生成する **prototype** ファイルにエントリーを追加する必要があることに注意してください。少なくとも、**pkginfo** ファイルのエントリーが必要です。詳細は **pkginfo** を参照してください。パッケージ内で使用するファイル (**copyright**、**compver**、**depend**、**setinfo**、**space**、パッケージ用に定義するすべてのインストール・スクリプトまたは除去スクリプト、定義するすべてのクラス、(例: **postinstall**)) のいずれについても、エントリーが必要です。

## フラグ

<b>-i</b>	シンボリック・リンクを無視し、ftype=f (ファイル) と ftype=s (シンボリック・リンク) としてパスを記録します。
<b>-c Class</b>	すべてのパスのクラスを <i>Class</i> にマップします。
<i>Path1</i>	オブジェクトが配置されるディレクトリーのパス。
<i>Path2</i>	<i>Path1</i> への出力上の置き換えられるべきパス。

## 例

以下の例は、**pkgproto** の使用と、生成された出力の部分リストを示しています。

- ```
$ pkgproto /usr/bin=bin /usr/usr/bin=usrbin /etc=etc
f none bin/sed=/bin/sed 0775 bin bin
f none bin/sh=/bin/sh 0755 bin daemon
f none bin/sort=/bin/sort 0755 bin bin
d none etc/master.d 0755 root daemon
f none etc/master.d/kernel=/etc/master.d/kernel 0644 root daemon
f none etc/rc=/etc/rc 0744 root daemon
```
- ```
$ find / -type d -print | pkgproto
d none / 755 root root
d none /usr/bin 755 bin bin
d none /usr 755 root root
d none /usr/bin 775 bin bin
d none /etc 755 root root
d none /tmp 777 root root
```
- 上記の例と同じですが、後で **pkgmk** で処理するためにファイルに取り込まれた出力が付きます。必要な **pkginfo** ファイルに追加されるエントリー、例えば、ファイルの後に実行の可能性があるインストール後のスクリプトが正しいロケーションにコピーされます。

```
$ find / -type d -print | pkgproto >prototype
$ (edit the file to add entries for pkginfo and postinstall)
$ cat prototype
i pkginfo
i postinstall
d none / 755 root root
d none /usr/bin 755 bin bin
d none /usr 755 root root
d none /usr/bin 775 bin bin
d none /etc 755 root root
d none /tmp 777 root root
```

## リターン・コード

- 0 スクリプトの正常終了です。
- 1 致命的なエラーです。この時点でインストール・プロセスが終了します。

## ファイル

`/usr/sbin/pkgproto`

`pkgproto` コマンドが入っています。

## 関連情報

`pkgmk` コマンド、`pkgparam` コマンド、`pkgtrans` コマンド。

`pkginfo` ファイル。

---

## pkgrm コマンド

### 目的

システムからパッケージまたはセットを除去します。

### 構文

#### インストールされたソフトウェア・パッケージを除去する

```
pkgrm [ -n ] [ -a Admin ] [-P Path ] [ Pkginst1 [ Pkginst2 [ . . . ] ] ]
```

#### スプール・デバイスからソフトウェア・パッケージを除去する

```
pkgrm -s Spool [ Pkginst ]
```

### 説明

`pkgrm` は、既にインストールされた、あるいは部分的にインストールされたパッケージまたはセットを、システムから除去します。パッケージは、個別にインストールできる、関連するファイルおよび実行可能ファイルの集合です。セットはセット・インストール・パッケージ (SIP) と呼ばれる特殊目的のパッケージと、そのセットのメンバーである 1 つまたは複数のパッケージのコレクションからなります。

`pkgrm` は、コマンド行にリストされたすべてのパッケージがシステムに含まれているかどうかを検査します。リストされたパッケージのいずれかが存在しない場合には、システムは変更されません。つまり、リストされたどのパッケージも除去されません。

除去されるパッケージに他のパッケージが依存していないかどうかを判別するための検査も行われます。依存関係が存在する場合に行われるアクションは、`Admin` ファイルで定義されています (以下の `-a` フラグを参照してください)。

このコマンドのデフォルト状態は対話モードです。このモードでは、行われるアクションを管理者が確認できるように、処理中にプロンプト・メッセージが出されます。 `-n` フラグを使用すると、非対話式モードを要求することができます。

`-s` フラグを使用すると、スプールされたパッケージを除去するディレクトリーを指定できます。

## フラグ

- n** 非対話式モードを使用可能にします。対話の必要がある場合には、このコマンドは終了します。このフラグを使用するためには、コマンドを起動するときに少なくとも 1 つはパッケージ・インスタンスを指定する必要があります。
- a Admin** デフォルトの管理ファイルの代わりに使用される、インストール先の管理ファイル *Admin* を定義します。( *Admin* ファイルの形式については、**admin** ファイル・フォーマットを参照してください。) "none" というトークンを指定すると、*admin* ファイルの使用がオーバーライドされ、ユーザーとの対話が強制されます。絶対パス名を指定しない限り、**pkgrm** は */var/sadm/install/admin* ディレクトリーからファイルを探します。デフォルトでは、そのディレクトリー内のファイル **default** が使用されます。
- P Path** 指定されたパッケージを、代替ルート・ディレクトリー・パスから除去します。
- s Spool** 指定されたパッケージをディレクトリー *Spool* から除去します。
- Pkginst** パッケージ/セット名の省略形を指定するために使用する短文字列を定義します。(「パッケージ・インスタンス」という用語は、厳密な意味で使用されているのではなく、インスタンス ID が指定されていないものも含め、*Pkginst* のすべてのインスタンス化を指します。)

*Pkginst* で SIP を指定すると、セットのメンバーおよび SIP 自体であるすべてのインストール済みパッケージが、依存関係と逆の順序で除去されます。

パッケージのすべてのインスタンスを指示するには、シェルが "\*" 文字を解釈しないようにするために、単一引用符でコマンド行を囲んで '*Pkginst.\**' と指定してください。ソース・メディアで使用可能なすべてのパッケージを参照するには、"all" というトークンを使用してください。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0** スクリプトの正常終了です。
- 1** 致命的なエラーです。この時点でインストール・プロセスが終了します。
- 99** 内部エラー

## ファイル

*/usr/sbin/pkgrm* **pkgrm** コマンドが入っています。

## 関連情報

**pkgadd** コマンド、**pkgask** コマンド、**pkgchk** コマンド、**pkginfo** コマンド、**pkgmk** コマンド、**pkgparam** コマンド、**pkgtrans** コマンド。

**pkginfo** ファイル・フォーマット、**pkgmap** ファイル・フォーマット。

---

## pkgtrans コマンド

### 目的

パッケージのフォーマットを変換します。

## 構文

**pkgtrans** [ **-i -o -n -s**] [ **-z Blocksize**] *Device1* *Device2* [ *Pkginst1* [ *Pkginst2* [...]]]

## 説明

**pkgtrans** コマンドは、あるフォーマットから別のフォーマットにインストール可能パッケージを変換します。以下の変換を行います。

- ファイルシステム・フォーマットからデータ・ストリームへ
- データ・ストリームからファイルシステム・フォーマットへ

**cs**h からは **pkgtrans** を実行できません。

## フラグ

- i** **pkginfo** ファイルと *Pkgmap* ファイルをコピーします。パッケージ・カテゴリがセット・インストール・パッケージ (SIP) を表す "set" として定義されている場合 (**setinfo** ファイル・フォーマットを参照)、そのパッケージの **setinfo** ファイルもコピーされます。
- o** 宛先デバイスにある同一インスタンスを上書きします。パッケージ・インスタンスが既に存在している場合には、上書きされます。
- n** パッケージの新規インスタンスを宛先デバイス上に作成します。宛先デバイスに既にそのパッケージ・インスタンスが存在する場合、そのインスタンスは変更されず、新規インスタンスが作成されます。新規インスタンスには、既存のインスタンスと区別するためのシーケンス番号が付加されます。例えば、宛先デバイスに既にパッケージ *X* のインスタンスが含まれているものとします。**-n** フラグを指定した **pkgtrans** を使用して、パッケージ *X* の新規インスタンスを宛先デバイスに書き込もうとすると、パッケージ *X* の既存インスタンスは宛先デバイスに残され、*X.2* という新規インスタンスがデバイスに作成されます。**-n** フラグを指定して **pkgtrans** を再度実行すると、*X.3* という 3 番目のインスタンスが作成されます。
- s** このパッケージを、ファイルシステムとしてではなくデータ・ストリームとして *Device2* に書き込むように指示します。デフォルトの振る舞いでは、*Device2* がファイルシステム・フォーマットで書き込まれます。
- z Blocksize** カートリッジ・テープに転送するときにブロック・サイズを使用するように指示します。**-z** フラグに 512 以外の値を指定してテープに書き込まれたパッケージは、常にブロック・サイズとして 32768 を使用して読み取られます。したがって、カートリッジ・テープから読み取る場合には、**-z** フラグは適用できません。
- Device1* ソース・デバイスを指定します。これは - (ハイフン) にすることができ、これはデータ・ストリーム・フォーマットのパッケージを標準入力から読み取ることを指定します。このデバイスの 1 つまたは複数のパッケージが変換されて、*Device2* に収められます。*Device1* が通常のファイルまたはディレクトリーである場合、相対パス名ではなく絶対パス名を使用しなければなりません。
- Device2* 宛先デバイスを指定します。これは - (ハイフン) にすることができ、これはパッケージをデータ・ストリーム・フォーマットで標準出力に書き込むことを指定します。変換されたパッケージはこのデバイスに入ります。*Device2* が通常のファイルまたはディレクトリーである場合、相対パス名ではなく絶対パス名としてこれを指定しなければなりません。
- Pkginst* *Device1* のどのパッケージを変換するのかを指定します。"all" というトークンを使用すると、すべてのパッケージを表すことができます。*Pkginst.\** を使用すると、あるパッケージのすべてのインスタンスを表すことができます。パッケージが定義されていない場合、プロンプトによってデバイス内のすべてのパッケージが示され、どれを変換するのか尋ねられます。あるセットをデータ・ストリーム形式に変換して転送する場合、*Pkginst* の引数として最初に SIP を指定し、その後で、その SIP の **setinfo** ファイルでリストされているパッケージを、リストされているとおりの順序で指定する必要があります。

注: 宛先デバイスに該当パッケージのいずれかのインスタンスが既に存在する場合、デフォルトでは、**pkgtrans** はそのパッケージのどのインスタンスも転送しません。 **-n** フラグを使用すると、このパッケージのインスタンスが既に存在する場合、新規インスタンスが作成されます。 **-o** フラグを使用すると、既に存在する同じインスタンスが上書きされます。宛先デバイスがデータ・ストリームである場合には、いずれにせよデータ・ストリーム全体が上書きされるため、これらのフラグはどれも有効ではありません。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0 スクリプトの正常終了です。
- 1 致命的なエラーです。この時点でインストール・プロセスが終了します。

## ファイル

**/usr/sbin/pkgtrans** **pkgtrans** コマンドが入っています。

## 例

- ドライブ *Device* にあるすべてのパッケージを変換し、変換したパッケージを **/tmp** に入れたい場合には、次のように入力してください。

```
pkgtrans Device /tmp all
```

- tmp** 内の "pkg1" と "pkg2" を変換し、*Device* にデータ・ストリーム形式で入れたい場合には、次のように入力してください。

```
pkgtrans -s /tmp Device pkg1 pkg2
```

## 関連情報

**pkgadd** コマンド、**pkgask** コマンド、**pkginfo** コマンド、**pkgmk** コマンド、**pkgparam** コマンド、**pkgrm** コマンド。

**pkginfo** ファイル・フォーマット。

---

## platform\_dump コマンド

### 目的

プラットフォーム (ハードウェアおよびファームウェア) ダンプ関連アクションを実行します。

### 構文

```
platform_dump { -c | -d | -e | -f fstype | -F flag | -l | -q | -S | -s seq_no } [ -L ]
```

### 説明

**platform\_dump** コマンドは、POWER5 プラットフォームに対するハードウェアおよびファームウェアの問題判別をサポートするために、AIX で初めて導入されました。このコマンドを使用すれば、オペレーティング・システムがファームウェア関連およびハードウェア関連のダンプを保管できるようになります。このコマンドは、保守権限が使用可能にされたパーティションでのみサポートされます (Hardware

Management Console (HMC) 管理対象システムは除外)。HMC 管理対象システムでは、このダンプは HMC に送られます。 **platform\_dump** コマンドは、一般に、オペレーティング・システム機能 (基本インストールおよび `dumpcheck` など) により実行されます。プラットフォーム・ダンプには以下のものが含まれます。

- ハードウェア状態
- hypervisor 状態
- FSP (Flexible Service Processor) 状態情報

プラットフォーム・ダンプ・ファイル用のディスク・スペースは、**platform\_dump** コマンドを使用して予約します。専用の論理ボリューム、`/dev/fwdump`、は `rootvg` ボリューム・グループ内に作成され、`/var/adm/ras/platform` ディレクトリ上にマウントされます。**fwdump\_dev** デバイスおよび **fwdump\_dir** マウント・ポイントの両方は、`SWservAt` オブジェクト・クラスの ODM に保管されます。インストールの間に、AIX は、**platform\_dump** コマンドを使用して、必要なディスク・スペースを予約します。ディスク・スペースが予約されるのは、パーティションがサービス・パーティションとして指定されている場合に限られます。プラットフォーム・ダンプの最大可能サイズは AIX に指示されるため、十分なスペースを事前にプラットフォーム・ダンプ用として割り当てることができます。このサイズは動的に変更されることがあることに注意してください。オペレーティング・システムは、これを検出すると、追加の所要量をユーザーに知らせ、可能であれば、論理ボリュームを自動的に拡張します。

**注:** AIX パーティションがインストールされた後このパーティションにサービス・パーティション権限を割り当てる場合は、**platform\_dump -f <fstype>** コマンドを実行して、`/dev/fwdump rootvg` 論理ボリュームを作成します。*fstype* 引数の持ちえる値は、**jfs2** または **jfs** 値です。

コマンド出力をエラー・ログに記録するために、**-L** フラグが提供されます。

## フラグ

- c** 見積もられたプラットフォーム・ダンプ・サイズ (ファームウェアによって示される)、およびプラットフォーム・ダンプ用に割り当てられたディスク・スペースの検査を実行します。次のように報告されます。見積もられたサイズが割り当てられたスペースより小さいか等しい場合は 0 を戻します。見積もられたサイズが割り当てられたスペースよりも大きい場合は、1 を戻します。
- d** プラットフォーム・ダンプ用に予約されたファイルシステム・スペースを削除し、他の使用目的のために解放します。予約済みディスク・スペースに既存のダンプ・ファイルがある場合は、失われることになります。
- e** プラットフォーム・ダンプが発生した場合にこれを保管するための、ディスク・スペースの見積もりを提供します。このオプションは、この見積もりを提供するためにファームウェアと対話します。ユーザーは、このスペース情報を基に、プラットフォーム・ダンプを保管するために十分なディスク・スペースを割り当てる必要があります。出力される値は、バイト単位での必要サイズになります。
- f *fstype*** プラットフォーム・ダンプ用に、システム上に十分なディスク・スペースを予約します。**-f** オプションは、プラットフォーム・ダンプ専用のファイルシステムが存在していなければ、これを作成します。既にファイルシステムが存在し、サイズが十分ではない場合は、ファイルシステム・サイズが増やされます。*fstype* は有効なファイルシステム・タイプであることが必要です。ファイルシステムが既に存在する場合、`any` を指定することができます。
- F *flag*** プラットフォーム・ダンプを使用可能または使用不可にします。フラグが 0 であればプラットフォーム・ダンプは使用不可であり、1 であればプラットフォーム・ダンプは使用可能です。
- l** プラットフォーム・ダンプの現行構成をリストします。



- L** **platform\_dump** に、出力を表示するとともにログに記録するよう伝えます。これは **-e** オプションによるサイズ出力には適用されません。
- q** プラットフォームがプラットフォーム・ダンプをサポートするかどうか、検査します。プラットフォーム・ダンプがサポートされる場合は 0 が戻されます。
- s seq\_no** ダンプ通知イベントで識別されたものと同じファームウェアから、プラットフォーム・ダンプを保管します。seq\_no は、AIX エラー・ログ・ファイル内に保管されているようなダンプ通知イベントのシーケンス番号を示します。このシーケンス番号は、詳細なデータ領域を解析し、ファームウェアからダンプ・データを入手するのに必要なダンプ・タグとダンプ情報を入手するために、このコマンドによって使用されます。
- S** データのスキャンをサポートするシステムに、スキャン・ダンプを保管します。このオプションが指定される場合、コマンドはスキャン・ダンプが存在するか検査します。存在していれば、既存のスキャン・ダンプ・インターフェースを使用してファームウェアから **scandump** データを読み取って保管します。

## 終了状況

- 0 正常終了。
- 1 **-c** が指定され、プラットフォーム・ダンプを保管するスペースが不足している場合に、戻されます。
- 255 プラットフォーム・ダンプがシステムでサポートされない場合に戻されます。
- 3 プラットフォーム・ダンプが使用不可である場合に戻されます。
- 2 エラーが発生した時に戻されます。

## セキュリティ

**platform\_dump** を実行できるのは root ユーザーのみです。

## 例

1. プラットフォーム・ダンプ・サイズの見積もりを入手するには、次のように入力します。

```
platform_dump -e
```

これにより、見積もられたプラットフォーム・ダンプ・サイズがバイト単位で報告されます。

## 関連情報

**dumpcheck** コマンド。

## plotgbe コマンド

### 目的

HP-GL ファイルをプロッター・デバイスにプロットします。

### 構文

```
/usr/lpd/plotgbe [ -fr=X ] [ -noin ] File
```



## 説明

**plotgbe** コマンドは、HP-GL ファイルをプロッター・デバイスにプロットするバックエンド・プログラムです。プロッター・デバイスは、5080 接続アダプターを使って 5085/5086 ワークステーションに接続しなければなりません。 **plotgbe** コマンドを使用するには、**plotgbe** バックエンド・プログラムの印刷キューを定義しなければなりません。 **SMIT** コマンドを使用して **plotgbe** バックエンドを指定するプロッター・キューを追加する方法は、 [How to Add Plotter Support with 5080](#) を参照してください。

**plotgbe** コマンドは、**qdaemon** プロセスによって呼び出されます。このコマンドはコマンド行に入力してはなりません。プロッターに対する特定の印刷要求に必要なオプションは、印刷ジョブを要求するのに使うコマンドとともに **plotgbe** コマンドに渡さなければなりません。 **enq** コマンドの場合は、**-o** フラグを使用して、処理のためにオプションを **plotgbe** バックエンドに渡します。

**plotgbe** バックエンド・コマンドは、プロッターの初期化とプロットのスケーリングのために適切な HP-GL コマンドも生成します。このデータがプロッターへ送信されてから、ユーザー指定の HP-GL ファイルが送信されます。つまり、HP-GL ファイルに含まれるスケーリングや初期化のコマンドは、**plotgbe** バックエンド・コマンドによって作成されるコマンドを無効にします。

**注:** ユーザーは、印刷要求コマンドとともに **plotgbe** コマンドに送信されたファイルに対して読み取りアクセスを持たなければなりません。

## フラグ

- fr=X** 複数フレームの図面のプロットを提供します。このオプションでは *X* 個のフレームがプロットされますが、この *X* は 1 から 9 までの範囲のいずれかの数字です。例えば、E サイズのロール・メディアに 20 フィートの図面をプロットするには、5 フレームが必要です。つまり、オプション **fr=5** が **plotgbe** バックエンドに渡されることになります。
- noin** デフォルト値にリセットしなくても、プロッターの正面パネル設定が現在のプロットにそのまま有効であるようにします。通常、プロッターにプロット・ページを定義する P1 と P2 の位置は、**plotgbe** コマンドによってそのデフォルトの位置に設定されます。 **-noin** 初期化なしオプションを使用して、デフォルトの位置を無効にします。

## 例

1. ファイル `longaxis.gl` を `plt` プロッター・キューに送信して、そのファイルが印刷に 5 つのフレームを必要とすることをバックエンドに指定するには、以下のように入力します。

```
enq -Pplt -o -fr=5 longaxis.gl
```
2. ファイル `plotdata.gl` を `plt` プロッター・キューに送信し、プロット・ページの位置がこのファイルのデフォルトにリセットされないように指定するには、以下のように入力します。

```
enq -Pplt -o -noin plotdata.gl
```
3. ファイル `twoplot.gl` を `plt` プロッター・キューに送信して、プロット・ページの初期化を行わず、そのプロッターが図面を 2 つのフレームで印刷するように指定するには、以下のように入力します。

```
enq -Pplt -o -noin -o fr=2 twoplot.gl
```

## ファイル

`/usr/lpd/plotgbe` **plotgbe** コマンドが入っています。

## 関連情報

**enq** コマンド、**plotlbe** コマンド、**qdaemon** コマンド。

プリンターおよび印刷 の 印刷の管理 のセクション。

プリンターおよび印刷 の 5080 によるプロッター・サポートの追加 のセクション。

---

## plotlbe コマンド

### 目的

HP-GL ファイルをプロッター・デバイスにプロットします。

### 構文

```
/usr/lpd/plotlbe [ -fr=X ] [ -noin ] File
```

### 説明

**plotlbe** コマンドは、TTY デバイスとして定義されるシリアル・ポートに接続したプロッターに HP-GL ファイルをプロットするバックエンド・プログラムです。 **plotlbe** コマンドを使用するには、シリアル・ポートの TTY デバイスを定義し、**plotlbe** バックエンド・プログラムの印刷キューを定義しなければなりません。

TTY シリアル・ポートを構成するときは、ボー・レート、パリティ、およびストップ・ビットをプロッターに適切に設定します。さらに、ユーザーの TTY ポートに対して XON/XOFF を FALSE に設定しなければなりません。

**plotlbe** コマンドは、**qdaemon** プロセスによって呼び出されます。このコマンドはコマンド行に入力してはなりません。プロッターに対する特定の印刷要求に必要なオプションは、印刷ジョブを要求するのに使うコマンド (通常は **enq** コマンド) とともに **plotlbe** コマンドに渡す必要があります。 **enq** コマンドの場合には、**-o** フラグを使用して、処理のためにオプションを **plotlbe** バックエンドに渡します。

**plotlbe** バックエンド・コマンドは、以下のプロッターをサポートします。すなわち、7731、7372、7374、7375-1、7375-2、6180、6182、6184、6186-1、および 6186-2 です。

**plotlbe** コマンドは、ENQ/ACK 初期接続手順をサポートします。初期接続手順の詳細は、ユーザーのプロッター・プログラミング解説書を参照してください。

**plotlbe** バックエンド・コマンドはプロッターの初期化とプロットのスケーリングのために適切な HP-GL コマンドも生成します。このデータがプロッターへ送信されてから、ユーザー指定の HP-GL ファイルが送信されます。つまり、HP-GL ファイルに含まれるスケーリングや初期化のコマンドは、**plotlbe** バックエンド・コマンドによって生成されるコマンドを無効にします。

**注:** ユーザーは、印刷要求コマンドとともに **plotlbe** コマンドに送信されるファイルに対して読み取りアクセスを持たなければなりません。

## フラグ

- fr=X** 複数フレームの図面のプロットを提供します。このオプションでは、*X* 個のフレームがプロットされますが、*X* は 1 から 9 までの範囲の数字です。例えば、E サイズのロール・メディアに 20 フィートの図面をプロットするには、5 つのフレームが必要です。つまり、オプション **-fr=5** は **plotlbe** バックエンドに渡されることとなります。
- noin** デフォルト値にリセットしなくても、プロッターの正面パネル設定が現在のプロットにそのまま有効であるようにします。通常、プロッターにプロット・ページを定義する P1 と P2 の位置は、**plotlbe** コマンドによってデフォルトの位置に設定されます。**-noin** 初期化なしオプションを使用して、デフォルトの位置を無効にします。

## 例

1. ファイル `longaxis.gl` を `plt` プロッター・キューに送信して、そのファイルが作図に 5 つのフレームを必要とすることをバックエンドに指定するには、以下のように入力します。

```
enq -Pplt -o -fr=5 longaxis.gl
```

2. ファイル `plotdata.gl` を `plt` プロッター・キューに送信し、プロット・ページの位置がこのファイルのデフォルトにリセットされないように指定するには、以下のように入力します。

```
enq -Pplt -o -noin plotdata.gl
```

3. ファイル `twoplot.gl` を `plt` プロッター・キューに送信して、プロット・ページの初期化を行わず、そのプロッターが図面を 2 つのフレームで印刷するように指定するには、以下のように入力します。

```
enq -Pplt -o -noin -o fr=2 twoplot.gl
```

## ファイル

`/usr/lpd/plotlbe` **plotlbe** コマンドが入っています。

## 関連情報

**enq** コマンド、**plotgbe** コマンド。

プリンターおよび印刷 の 印刷の管理のセクション。

---

## pmcycles コマンド

### 目的

プロセッサのクロック速度を測ります。

### 構文

```
pmcycles [ -d] [ -m]
```

### 説明

**pmcycles** コマンドは、パフォーマンス・モニターのサイクル・カウンターとプロセッサ・リアルタイム・クロックを使用して、実際のプロセッサ・クロック速度を MHz 単位で測ります。また、オプションにより、速度の減分を MHz 単位およびティックあたりのナノ秒数で示します。減分係数は、クロック

がゼロになるたびにクロック割り込みを生成する、2 進カウンターです。ティックは減分の値です。マシンによって、時間がナノ秒単位で減分するために各ティックが 1 ナノ秒に等しくなることも、また、減分の値がマシンに依存することもあります。

このコマンドは、**bos.pmapi** によってサポートされるプロセッサだけでサポートされます。

## フラグ

**-d** 速度の減分を MHz 単位およびティックあたりのナノ秒数で示します。  
**-m** それぞれのプロセッサの速度を表示します。

## 例

1. プロセッサ速度を表示するには、次のように入力します。

```
pmcycles
```

下記のような出力が表示されます。

```
This machine runs at 133 MHz
```

2. それぞれのプロセッサ速度を表示するには、次のように入力します。

```
pmcycles -m
```

下記のような出力が表示されます。

```
Cpu 0 runs at 200 MHz  
CPU 1 runs at 200 MHz
```

## 関連情報

pm\_cycles サブルーチン。

---

## pmllist コマンド

### 目的

サポートされているプロセッサについての情報を表示します。

### 構文

```
pmllist [ -h ]
```

```
pmllist -l [ -o t | c ]
```

```
pmllist { -s | -e ShortName | -c Counter [ ,event ] | -g Group | -S Set | -D DerivedMetricID } [ -p  
ProcessorType ] [ -s ] [ -d ] [ -o t | c ] [ -f Filter ]
```

### 説明

**pmllist** コマンドは以下の機能を実行します。

- サポートされているプロセッサのリストを表示します。
- 指定されたプロセッサについての情報の要約を表示します。
- 指定されたプロセッサのイベント・テーブルを表示します。

- 指定されたプロセッサの既存のイベント・グループを表示します。
- 指定されたプロセッサの既存のイベント・セットを表示します。
- イベント・セットおよび指定されて導出されたメトリックに対する数式を表示します。

## フラグ

<b>-c</b> <b>-1</b>	すべてのカウンターのすべてのイベントを表示します。
<b>-c</b> <i>Counter</i>	指定したカウンター <i>Counter</i> のすべてのイベントを表示します。
<b>-c</b> <i>Counter,Event</i>	指定したカウンター <i>Counter</i> の指定したイベント <i>Event</i> のリストを表示します。
<b>-d</b>	イベントの詳細記述を表示します。
<b>-D</b> <b>-1</b>	サポートされる派生メトリックをすべて表示します。
<b>-D</b> <i>DerivedMetricID</i>	指定された <i>DerivedMetricID</i> を表示します。
<b>-e</b> <i>ShortName</i>	すべてのカウンターの指定した <i>ShortName</i> の説明を表示します。
<b>-f</b> <i>v,u,c</i>	フィルターをコンマで区切ったリストとして、イベント・フィルターを指定します。有効なフィルターは、 <b>v</b> (検査済み)、 <b>u</b> (未検査)、および <b>c</b> (要注意) です。それらのフィルターは、イベントのテスト状況を表しています。デフォルトのフィルターは <b>v,u,c</b> です。
<b>-g</b> <b>-1</b>	すべてのイベント・グループのリストを表示します。
<b>-g</b> <i>Group</i>	指定したイベント・グループ <i>Group</i> を表示します。
<b>-h</b>	<b>pmlist</b> コマンドのヘルプ情報を表示します。
<b>-l</b>	サポートされているすべてのプロセッサ・タイプのリストを表示します。
<b>-o</b> <i>tic</i>	<b>pmlist</b> コマンドの出力形式を指定します。有効な出力形式は、 <b>t</b> (テキスト形式) および <b>c</b> (CSV 形式) のいずれかで指定されます。デフォルトの出力形式はテキストです。
<b>-p</b> <i>ProcessorType</i>	プロセッサ・タイプを指定します。
<b>-s</b>	プロセッサ情報の要約を表示します。
<b>-S</b> <b>-1</b>	サポートされるイベント・セットをすべて表示します。
<b>-S</b> <i>Set</i>	指定されたイベント <i>Set</i> を表示します。

## 例

1. サポートされているすべてのプロセッサを表示するには、次のように入力します。  
`pmlist -l`
2. 現在のプロセッサの要約情報を表示するには、次のように入力します。  
`pmlist -s`
3. 現在のプロセッサの要約情報を CSV 形式で表示するには、次のように入力します。  
`pmlist -s -o c`
4. 現在のプロセッサのグループ番号 62 を表示するには (現在のプロセッサでイベント・グループがサポートされている場合)、次のように入力します。  
`pmlist -g 62`
5. POWER4 プロセッサのカウンター 1 のイベント 3 の詳細情報を表示するには、次のように入力します。  
`pmlist -p POWER4 -c 1,3 -d`

- 現在のプロセッサのセット番号 2 を表示するには (現在のプロセッサでイベント・セットがサポートされている場合)、次のように入力します。

```
pmlist -S 2
```

## 関連情報

**pm\_initialize** サブルーチン。

---

## pmtu コマンド

### 目的

パス MTU ディスカバリー関連情報を表示して削除します。

### 構文

```
pmtu [-inet6] display/[delete [-dst destination] [-gw gateway] ]
```

### 説明

**pmtu** コマンドは、パス MTU 情報を管理するために提供されます。コマンドを使用して、パス MTU テーブルを表示することができます。デフォルトでは、Ipv4 **pmtu** エントリーが表示されます。Ipv6 **pmtu** エントリーは、**-inet6** フラグを使用して表示することができます。またこのコマンドにより、root ユーザーが **pmtu delete** コマンドで **pmtu** エントリーを削除することが可能になります。削除は、宛先、ゲートウェイ、またはその両方を基にして行うことができます。

**pmtu** エントリーは、MTU 値を指定して経路追加が発生した場合に、PMTU テーブルに追加されます。

ネットワーク・オプション **pmtu\_expire** は、使用されない **pmtu** エントリーを期限切れにするために提供されます。 **pmtu\_expire** のデフォルト値は 10 分です。

### フラグ

<b>-dst</b>	削除されるべき <b>pmtu</b> エントリーの宛先を指定します。
<b>-gw</b>	削除されるべき <b>pmtu</b> エントリーのゲートウェイを指定します。
<b>-inet6</b>	Ipv6 <b>pmtu</b> エントリーを表示するかそれとも削除するか、指定します。

### 終了状況

0	コマンドは正常に実行されました。
1	エラーが発生しました。

### 例

- Ipv4 **pmtu** エントリーを表示するには、次のように入力します。

```
pmtu display
```

出力は次のようになります。

```
dst          gw          If    pmtu    refcnt    redisc_t    exp
-----
192.168.5.5  192.168.10.33  en2   1500    1         0         0
```

参照カウントは、この `pmtu` エントリーを使用している現行 TCP アプリケーションおよび UDP アプリケーションの数を示します。

`redisc_t` エントリーは、最後のパス MTU ディスカバリーの試みからの経過時間を示します。PMTU は、`pmtu_rediscover_interval` の分数ごとに、再ディスカバリーされます。デフォルト値は 30 分で、`no` コマンドを使用して変更することができます。

PMTU エントリーの期限切れは、ネットワーク・オプション `pmtu_expire` によって制御されます。デフォルト値は 10 分です。この値は、`no` コマンドを使用して変更することができます。値 0 の場合、期限切れになるエントリーはありません。`exp` エントリーが有効期限時刻を示します。ゼロより大きい `refcnt` を持つ PMTU エントリーの `exp` は 0 です。`refcnt` がゼロになると、`exp` 時間は 1 分ごとに増え、`exp` 変数が `pmtu_expire` と等しくなったとき、エントリーは削除されます。

- 宛先を基にしてエントリーを削除するには、次のように入力します。

```
pmtu delete -dst 192.168.5.5
```

- `Ipv6` を表示するには、次のように入力します。

```
pmtu -inet6 display
```

出力は次のようになります。

```
dst          gw          If    pmtu    refcnt    redisc_t    exp
-----
fe80::204:acff:fee4:ab3b  ::    lo0    16896    2         2         0
```

## 位置

`/usr/sbin/pmtu`

## ファイル

`/usr/sbin/pmtu` `pmtu` コマンドが入っています。

## 関連情報

260 ページの『`no` コマンド』、868 ページの『`route` コマンド』。

## pop3d デーモン

### 目的

Post Office Protocol バージョン 3 (POP3) のサーバー・プロセスを始動します。

### 構文

```
pop3d [-c]
```



## 説明

**pop3d** コマンドは POP3 サーバーです。これは、POP3 リモート・メール・アクセス・プロトコルをサポートします。これはまた、標準入力コマンドを受け入れ、標準出力で応答します。**pop3d** コマンドは通常、リモート・クライアント接続に付加されたディスクリプターを指定した **inetd** デーモンとともに呼び出します。

**pop3d** コマンドは、**sendmail** および **bellmail** で構成された、既存のメール基礎構造とともに作動します。

## フラグ

**-c** ホスト名のリバース・ルックアップを抑制します。

## パラメーター

なし

## 終了状況

すべてのエラーおよび状況情報は、**syslogd** がログ記録に対して構成されている場合はログ・ファイルに書き込まれます。

## セキュリティ

**pop3d** デーモンは、**imap** というサービス名を持つ、PAM 使用可能アプリケーションです。認証に PAM を使用するシステム規模の構成は、**/etc/security/login.cfg** の **usw** スタンザにある **auth\_type** 属性の値を、**root** ユーザーと同じ **PAM\_AUTH** に変更することにより、設定されます。

PAM が使用可能であるときに使用される認証メカニズムは、**/etc/pam.conf** 内の **imap** サービスの構成によって決まります。**pop3d** デーモンは、**auth** モジュール・タイプおよび **session** モジュール・タイプの **/etc/pam.conf** エントリーを必要とします。以下にリストされるのは、**imap** サービスにおける **/etc/pam.conf** 内の推奨される構成です。

```
#  
# AIX imap configuration  
#  
imap auth      required    /usr/lib/security/pam_aix  
  
imap session   required    /usr/lib/security/pam_aix
```

注: **pop3d** デーモンは **imap** ライブラリーを認証に使用するので、**imap** サービスは **imapd** デーモンおよび **pop3d** デーモンの両方に使用されます。

## ファイル

**/usr/sbin/pop3d** **pop3d** コマンドが入っています。  
**/etc/services** 必要なサービスのポート割り当てでファイルを指定します。以下の入力はこのファイル内になければなりません。  
  
pop3 110/tcp postoffice3 # Post Office Protocol Ver. 3



```
#
# AIX imap configuration
#
imap auth      required    /usr/lib/security/pam_aix
imap session   required    /usr/lib/security/pam_aix
```

注: **pop3ds** デーモンは **imap** ライブラリーを認証に使用するので、**imap** サービスは **imapds** デーモンおよび **pop3ds** デーモンの両方に使用されます。

## ファイル

**/usr/sbin/pop3ds**            **pop3ds** コマンドが入っています。  
**/etc/services**            必要なサービスのポート割り当てでファイルを指定します。以下の入力はこのファイル内になければなりません。  
  
pop3s 995/tcp # pop3 protocol over TLS/SSL

## 関連情報

デーモン: **imapds**

---

## portmap デーモン

### 目的

RPC プログラム番号をインターネットのポート番号に変換します。

### 構文

**/usr/sbin/portmap**

### 説明

**portmap** デーモンは、RPC プログラム番号をインターネットのポート番号に変換します。

RPC サーバーは、始動時に **portmap** デーモンを使って登録します。サーバーはデーモンに対して、listen しているポート番号と使用する RPC プログラム番号を通知します。したがって、**portmap** デーモンは、ホスト上のすべての登録済みポートの位置と、これらの各ポート上で使用可能なプログラムがわかります。

クライアントは、呼び出そうとするプログラムごとに一度だけ、**portmap** デーモンに問い合わせます。**portmap** デーモンはクライアントに対して、その呼び出しの送信先となるポートを通知します。クライアントは、この情報を将来の参照のために格納します。

標準 RPC サーバーは通常 **inetd** デーモンによって始動されるので、**portmap** デーモンは **inetd** デーモンが呼び出される前に始動しなければなりません。

注: **portmap** デーモンが停止または異常終了した場合には、ホスト上にある RPC サーバーをすべて再始動しなければなりません。

## フラグ

なし

## 例

1. **portmap** デーモンを始動するには、以下のコマンドを入力します。

```
startsrc -s portmap
```

2. **portmap** デーモンを停止するには、以下のコマンドを入力します。

```
stopsrc -s portmap
```

## ファイル

**inetd.conf**

RPC デーモンおよびその他の TCP/IP デーモンを始動します。

**/etc/rpc**

サーバー名およびそれに対応する **rpc** プログラム番号と別名のリストが入っています。

## 関連情報

**rpcinfo** コマンド。

**inetd** デーモン。

*AIX 5L* バージョン 5.3 ナショナル・ランゲージ・サポート ガイドおよびリファレンスのネットワーク・ファイルシステム (NFS) の概要のセクション。

オペレーティング・システムおよびデバイスのマネージのシステム・リソース・コントローラーのセクション。

ネットワークおよびコミュニケーションのマネージ の TCP/IP プロトコルのセクション。

ネットワークおよびコミュニケーションのマネージ の NFS コマンドのセクション。

---

## portmir コマンド

### 目的

TTY ストリーム (モニター) が、別の TTY ストリーム (ターゲット) に接続して、そのストリームで発生しているユーザー・セッションをモニターできるようにします。

### 構文

```
portmir { -d mir_modem -t target [ -m monitor ] | -t target [ -m monitor ] | { -o | -c monitor | -q } }
```

### 説明

**portmir** コマンドによって、ある TTY ストリーム (モニター) が、ほかの TTY ストリーム (ターゲット) に接続して、そのストリームで起こっているユーザー・セッションをモニターすることができるようになります。これは、ターゲットおよびモニターの両方の TTY ストリームに特殊「ミラー」モジュールをプッシュすることによって実行されます。

ターゲットおよびモニターの両方の TTY は、モニターするセッションが開始したとき、それぞれのディスプレイに表示メッセージを受信します。モニターするセッションは、ターゲット TTY から、モニター TTY から、またはモニターするセッションに関係ない第 3 の TTY から終了することができます。サービス中でないモードでモニターが使用される場合は、コマンドが作動するために、両方のストリームはオープン状態になければなりません (すなわち、各 TTY で `getty` またはアクティブ・セッションのいずれか

が起こっていなければなりません)。これは、「ミラー」ストリーム・モジュールのプッシュを許可するために必要です。 **portmir** コマンドは、TTY デバイスのみ (PTS, TTY, LFT) での使用がサポートされています。

TERM 環境変数で定義されているように、端末タイプはモニターおよびターゲット TTY の両方に対して同じでなければなりません。この環境変数の値は、**terminfo** データベース内の有効な入力に対応しなければなりません。端末タイプの例としては、**ibm3151** または **vt100** があります。LFT は **vt100** と同様です。**aixterm** などの端末エミュレーターは通常、機能上 **vt100** と同様です。

コンソールはターゲット TTY またはモニター TTY として使用できますが、モニター TTY として使用することはお勧めしません。ただし、コンソールがモニター TTY として使用される場合は、コンソールは最初、モニターするセッションが持続している間は自動的にターゲット TTY に宛先変更されることに注意してください。モニターするセッションが終了したとき、コンソールは **CuAt ODM** データベース属性 **syscons** で指定されたデバイスに宛先が再変更されます。コンソールが以前に宛先変更されていた場合は、宛先変更は保存されません。

文字処理のオフロードを提供する非同期デバイスには、この機能を提供するために伝送制御手順 (**ldterm**) に依存するデバイスをミラーしている場合、問題がある可能性があります。この例としては、128-port 非同期アダプターがあります。異なるアダプターのポートがモニターされる場合、**chdev** コマンドを使用して、**fastcook** 属性を使用不可にします。コマンドは以下のように実行してください。

```
chdev -l tty1 -a fastcook -disable
```

Web-based System Manager (wsm) の デバイス アプリケーションを使用してデバイス特性を変更できます。

## フラグ

- c monitor** **device** パラメーターを値フィールドとして含む **CuAt ODM** データベース属性 **portmir\_monitor** を作成することによって、サービス・ブートにポートを構成します。このデバイスは後に、**portmir** コマンドがサービス・モード (**-s** フラグ) で呼び出されたとき、デフォルトのモニターするデバイスとして使用されます。  
  
ミラーリングはシステム管理担当者によって構成され、**-c** オプションを使用してサービス・ブート時に実行しなければなりません。ターゲットは **portmir\_monitor** 属性で定義されるデバイスにデフォルト設定されています。
- d mir\_modem** ダイアルインの目的でモニターするポートを設定します。root ユーザーだけが、このフラグでコマンドを発行することができます。**/usr/share/mir\_modem** が、正しいモデム・セットアップ・ファイルにリンクされていることを確認してください。**/usr/share/mir\_modem** には見本のファイルが入っています。モデムのタイプによって、自分自身のファイルを作成することも可能です。
- m monitor** モニターするデバイスを指定します。**-m** オプションと **-s** オプションのどちらも指定されない場合は、モニターするデバイスは **portmir** コマンドが実行されていたポートにデフォルト設定されています。
- o** モニターをオフにし、コマンドを終了します。
- q** **-c** オプションで値セットを照会します。
- t target** モニターされるターゲット・デバイスを指定します。

## セキュリティ

単一のミラー・セッションのみが任意の時に実行中である可能性があります。

ポートを非サービス・モードでミラーするには、ホーム・ディレクトリー内の **.mir** ファイル内にそれをモニターする可能性のあるユーザーのリストを配置します (root ユーザーには必要でない)。**mirror** デーモンが実行し始めると、デーモンは誰がそのポート上にいるかを検査します。また、モニターするポートのユーザーがそのポートをモニターする権利があるかどうかを検査します。

**.mir** ファイルは、ラインごとに単一のユーザー ID というフォーマットでなければなりません。

**重要:** **su** コマンドを実行してミラー・セッションの間に root ユーザーに変更すると、両方のユーザーに root 権限が与えられます。

## 例

1. **user1** が **user2** のログイン ID を **/u/user2/.mir** ファイルに配置した後、ターゲット **tty1** の **user1** をモニター **tty2** の **user2** からミラーするには、以下を入力してください。

```
portmir -t tty1 -m tty2
```

2. ターゲット **tty1** をモニター **tty2** のダイヤルインしているユーザーにミラーするには、以下を入力してください。

```
portmir -t tty1 -m tty2 -d mir_modem
```

3. サービス・ブートの間にモニターするデバイスを指定して、サービス・ブートに対してミラーリングを設定するには、以下を入力してください。

```
portmir -c tty
```

4. サービス・ブートの間にミラーリングを使用不可にするには、以下を入力してください。

```
portmir -c off
```

5. サービス・ブートのミラーリング・デバイスを照会するには、以下を入力してください。

```
portmir -q
```

## ファイル

**/usr/share/modems/mir\_modem**   ダイヤルインの設定のモデム構成ファイル例。  
**/usr/sbin/portmir**            コマンド・ファイルが入っています。

## 関連情報

**chdev** コマンド。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

---

## post コマンド

### 目的

メッセージの経路を指定します。

### 構文

```
post [ -alias File ... ] [ -format | -noformat ] [ -msgid | -nomsgid ] [ -filter File | -nofilter ]  
[ -width Number ] [ -verbose | -noverbose ] [ -watch | -nowatch ] File
```

## 説明

**post** コマンドは、メッセージを正しい宛先へ経路指定します。**post** コマンドは、ユーザーが始動することではできません。**post** コマンドは、ほかのプログラムによってのみコールすることができます。

**post** コマンドは、メッセージで受信側のアドレスを指定するすべてのコンポーネントを検索し、各アドレスを解析して正しいフォーマットかどうかを検査します。**post** はまた、アドレスを標準フォーマットにして、**sendmail** コマンドをコールします。**post** コマンドはさらに、Date: および From: コンポーネントを追加して、Bcc: コンポーネントを処理するなどの、ヘッダー操作を行います。**post** コマンドは *File* パラメーターを使用して、固定表示されるファイルの名前を指定します。

**注:** **post** コマンドは複雑なアドレス (例えば @A:harold@B.UUCP) を解析するときに、エラーを報告することがあります。複雑なアドレスを使用する場合には、**post** コマンドではなく **spost** コマンドを使用してください。

## フラグ

<b>-alias</b> <i>File</i>	指定されたメール別名ファイルでアドレスを検索します。このフラグを繰り返して、複数のメール別名ファイルを指定できます。 <b>post</b> コマンドは、自動的に <i>/etc/mh/MailAliases</i> ファイルを検索します。
<b>-filter</b> <i>File</i>	指定されたファイルでヘッダー・コンポーネントを使用して、Bcc: 受信側に送信されるメッセージをコピーします。
<b>-format</b>	送達トランスポート・システム用にすべての受信側アドレスを標準フォーマットにします。このフラグがデフォルトです。
<b>-help</b>	コマンド構文、使用可能なスイッチ (トグル)、およびバージョン情報をリストします。 <b>注:</b> メッセージ・ハンドラー (MH) の場合は、このフラグの名前は完全につづらなければなりません。
<b>-msgid</b>	メッセージにメッセージ識別コンポーネント (Message-ID: など) を追加します。
<b>-nofilter</b>	To: および cc: 受信側へのメッセージから、Bcc: ヘッダーを除去します。メッセージに最小ヘッダーを付けて、Bcc: 受信側に送信します。このフラグがデフォルトです。
<b>-noformat</b>	受信側のアドレスのフォーマットを変更しません。
<b>-nomsgid</b>	メッセージにメッセージ識別コンポーネントを追加しません。このフラグがデフォルトです。
<b>-noverbose</b>	メッセージを <b>sendmail</b> コマンドに送達する間は情報を表示しません。このフラグがデフォルトです。
<b>-nowatch</b>	<b>sendmail</b> コマンドによる送達の間は情報を表示しません。このフラグがデフォルトです。
<b>-verbose</b>	メッセージを <b>sendmail</b> コマンドに送達する間に情報を表示します。この情報によって、実行されているステップをモニターすることができます。
<b>-watch</b>	メッセージを <b>sendmail</b> コマンドによって送達する間に情報を表示します。この情報によって、実行されているステップをモニターすることができます。
<b>-width</b> <i>Number</i>	アドレスが入っているコンポーネントの幅を設定します。デフォルトは 72 桁です。

## ファイル

<i>/etc/mh/MailAliases</i>	デフォルトのメール別名が入っています。
<i>/etc/mh/mtstailor</i>	MH コマンド定義が入っています。



## 関連情報

**ali** コマンド、 **conflict** コマンド、 **mhmail** コマンド、 **send** コマンド、 **sendmail** コマンド、 **spost** コマンド、 **whom** コマンド。

**.mh\_alias** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ のメール・アプリケーションのセクション。

---

## pppattachd デーモン

### 目的

非同期デバイス・ストリームを PPP (Point-to-Point プロトコル) サブシステムに接続します。デーモン・プロセスまたは通常プロセスとして起動できます。

### 構文

特定の tty ポートを接続として使用する (デーモンとして実行する) には

```
pppattachd /dev/ttyPortNumber { client | server | demand } { ip | ipv6 | ip ipv6 } [ multilink ] [ connect "ConnectorProgram" ] [ inactive Seconds ] [ authenticate pap | chap ] [ peer pap | chap ] [ user Name ] [ remote HostName ] [ nodaemon ]
```

標準入力および標準出力を tty デバイスとして使用する (プロセスとして実行する) には

```
pppattachd { client | server | demand } { ip | ipv6 | ip ipv6 } [ multilink ] [ inactive Seconds ] [ authenticate pap | chap ] [ peer pap | chap ] [ user Name ] [ remote HostName ] [ nodaemon ]
```

### 説明

**pppattachd** デーモンは、非同期ストリームを PPP サブシステムに結合するためのメカニズムを提供します。発信接続を特定の tty ポートに配置すると、**pppattachd** はデーモンになります。stdin (標準入力) および stdout (標準出力) を PPP コミュニケーションに対する tty デバイスとして使用すると、**pppattachd** はデーモンになりません。(これは tty デバイスのログイン時に **\$HOME/.profile** から実行されます。)

PAP または CHAP 認証は、**authenticate** および **peer** オプションで活動化することができます。**smit** コマンドを使用して、**/etc/ppp/pap-secrets** または **/etc/ppp/chap-secrets** ファイルに入力を作成してください。**pppattachd** デーモンは、これらのファイルでパスワードを使用し、接続を認証します。これは、**/etc/ppp/pap-secrets** ファイルのみで PAP 認証を検索し、**/etc/ppp/chap-secrets** ファイルで CHAP 認証を検索します。

マルチリンク・オプションは、PPP リンクを、2 つの PPP ピア間で複数のアタッチメントを持つものとして識別するために使用します。PPP パケットは 1 つのピアでフラグメント化され、複数のアタッチメントに送信され、マルチリンクもサポートしなければならないリモート・ピアに再接続されます。最大受信再構成単位 (MRRU) および端点ディスクリプターは、PPP リンク構成メニューの **SMIT** を通して設定されます。MRRU はフラグメント化の前の最大データ・サイズです。端点判別プログラムは、一意的にローカル・システムを識別します。

エラーおよびメッセージは **syslog** 機能を使用してログに記録されます。

## オプション

**authenticate pap | chap**

**client | server | demand**

**ip | ipv6 | ip ipv6**

**connect "ConnectorProgram"**

**inactive** *Seconds*

**multilink**

**nodaemon**

**peer pap | chap**

**remote** *HostName*

**user** *Name*

現行システムを、PAP または CHAP のいずれかの認証ホストとして定義します。

サブシステム接続のタイプを、デーモンが実行中のシステム上で結合されるように定義します。

プロトコル・タイプを指定します。

発信接続を配置するために使用するプログラムを指定します。開かれた tty デバイスは、stdin および stdout としてプログラムに渡されます。

**/usr/sbin/pppdial** コマンドは、使用できるコネクタ・プログラムです。接続終了の前に、リンク上で非活動を待機する秒数 (符号なし整数) を指定します。デフォルト値は 0 (タイムアウトなし) です。

PPP リンクを、2 つの PPP ピアに接続するアタッチメントのグループを持つものとして識別します。

接続プロセスに、デーモンにならないように指定します。このオプションは、要求接続で呼び出される接続プロセスに使用しなければなりません。

現行システムを、PAP または CHAP のいずれかのピアとして定義します。

リモート・ホスト名が、PAP 認証に使用されるように定義します。正常に接続するには、*UserName RemoteHostName Password* に対する入力、

**/etc/ppp/pap-secrets** ファイルに存在しなければなりません。このオプションは、認証およびピアの両方で PAP 認証に対して意味を持つだけです。

ユーザー入力を PAP 認証に対して使用するよう定義します。正常に接続するには、*UserName RemoteHostName Password* に対する入力、

**/etc/ppp/pap-secrets** ファイルに存在しなければなりません。このオプションは、ピアで PAP 認証に対する意味を持つだけです。

## 終了状況

このコマンドは、以下の終了値を戻します。

0 正常終了。  
>0 エラーが発生しました。

## セキュリティ

アクセス制御: 任意のユーザー

イベントの監査: N/A

## 例

1. システム A が、サーバー・システム B に対するクライアントとして機能するようにしたい場合は、システム A から以下のように入力します。

```
/usr/sbin/pppattachd /dev/tty0 client ip connect "sysbconnector"
```

ここで、sysbconnector はコネクタ・プログラムです。

システム B では、ログインしたユーザーは **\$HOME/.profile** から以下を呼び出しました。

```
exec /usr/sbin/pppattachd server ip 2>/dev/null
```

2. サーバー・システム B が、クライアント・システム A にコンタクト可能する場合は、システム B から以下のように入力します。

```
/usr/sbin/pppattachd /dev/tty0 server ipv6 connect "sysaconnector"
```

ここで、sysaconnector はコネクタ・プログラムです。

システム A では、ログインしたユーザーは **\$HOME/.profile** から以下を呼び出しました。

```
exec /usr/sbin/pppattachd client ipv6 2>/dev/null
```

3. PAP 認証を使用して、システム A をサーバー・システム B に対するクライアントとして機能させたいとします。システム B は認証ホストとして作動し、システム A は認証されるピアです。システム A から以下のように入力します。

```
/usr/sbin/pppattachd /dev/tty0 client ip ipv6 peer pap user username \  
connect "sysbconnector"
```

ここで、sysbconnector はコネクタ・プログラムです。

システム A では、**/etc/ppp/pap-secrets** ファイルには username \* ppppassword が入っています。システム B では、ログインしたユーザーは **\$HOME/.profile** から以下を呼び出しました。

```
exec /usr/sbin/pppattachd server ip ipv6 authenticate pap 2>/dev/null
```

システム B では、**/etc/ppp/pap-secrets** ファイルには username \* ppppassword が入っています。

## ファイル

**/usr/sbin/pppattachd**  
**/etc/ppp/attXXX.pid**

**pppattachd** デーモンが入っています。  
プロセス ID が入っています。XXX は pid であり、ファイルの内容はアタッチメントがバインドされるネットワーク・レイヤー ID です。pid ファイルが作成されるためには、ユーザーは uucp グループに属さなければなりません。

## 関連情報

**pppcontrold** デーモン、**pppdial** コマンド。

**syslog** サブルーチン。

**profile** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ の非同期 PPP サブシステムのセクション。

---

## pppcontrold デーモン

### 目的

PPP (Point-to-Point プロトコル) サブシステムの始動と管理を制御します。

### 構文

システム・リソース・コントローラーを使用して始動および停止するには

```
startsrc -s pppcontrold
```

```
stopsrc -s pppcontrold
```

## 説明

**pppcontrold** デーモンは、PPP サブシステムのインストールおよび構成のために **/etc/ppp/lcp\_config** および **/etc/ppp/if\_conf** ファイルを読み込みます。**/etc/ppp/lcp\_config** および **/etc/ppp/if\_conf** の両方を生成するには、**SMIT** を使用すべきです。これらのファイルを変更するには、ユーザーは **root** 権限を持っているか、または **uucp** グループのメンバーである必要があります。構成ファイルは、適切なストリーム・モジュールが構成されロードされる初期化の際に読まれ、**tcpip** ネットワーク・インターフェース・レイヤーはシステムにインストールされます。サブシステムの構成後、**pppcontrold** デーモンは **IP** アドレス、および **IP** および **IPv6** インターフェースのフラグの設定などの操作を行うために、**IP** および **IPv6** インターフェースに関連したストリームをモニターします。**pppcontrold** デーモンは、**SIGTERM** の受信時または **stopsrc** コマンドの起動時に終了します。**pppcontrold** デーモンの始動および停止にお勧めするメソッドは、**SRC** (システム・リソース・コントローラー) を使用するものです。**src** コマンドを実行するには、**root** 権限を持っていないければなりません。

エラーおよびメッセージは **syslog** 機能を使用してログに記録されます。

**pppcontrold** デーモンは、**pppcontrold** デーモンを終了するために使用するコマンド・プロセス ID のある単一行を含む **/etc/ppp/pppcontrold.pid** ファイルを作成します。

## フラグ

なし

### **/etc/ppp/lcp\_config** ファイル

このファイルは、サブシステムに必要な構成情報を提供します。これらの値は、サブシステムが構成される時のストレージの正しい割り当てを保証するために使用されます。これらの値はカーネル内で割り当てられるストレージを定義するため、必要なものだけを構成することが大切です。構成ファイルでは、ブランク行および **#** (ポンド記号) で始まる行は無視されます。インターフェース定義内で、ブランク行および **#** (ポンド記号) で始まる行を使用しないでください。これらの行は、インターフェース定義間のみで使用してください。

#### 必要なキーワード

**server\_name** *name*

このシステムの名前。この名前はシステムに固有でなければなりません。名前の初めの 20 バイトが固有であるようにしてください。

**lcp\_server** #

サーバー接続の数。サブシステムに許可されるサーバー接続の数を表します。指定されたすべての接続に対するストレージは、サブシステムが構成されるときに割り当てられます。最小値は 0 で、最大値はシステム内のメモリー量によってゲートされます。

**lcp\_demand** #

PPP LCP マルチプレクサーにサポートさせる要求リンクの最大数を指定します。この値は、構成する予定の要求インターフェースの数に設定します。デフォルト値は 0 です。

**lcp\_client** #

クライアント接続の数。最小値は 0 で、最大値はシステム内のメモリー量によってゲートされます。クライアント接続は、アドレスなしで構成される **IP** および **IPv6** インターフェースです。

**num\_if** #

構成する **IP** および **IPv6** インターフェースの数。**lcp\_server** + **lcp\_client** 以下でなければなりません。

## 必要なキーワード

**num\_if6 #**

許容する TCP/IPv6 インターフェースの最大数。この値は 10 進数です。この数値を「max ip interfaces (IP インターフェースの最大数)」および「max ip & ipv6 interfaces (IP および IPv6 インターフェースの最大数)」と併せた値が、サーバー・リンク、クライアント・リンク、および要求リンクの最大数の合計を上回ってはいけません (max server links (サーバー・リンクの最大数) + max client links (クライアント・リンクの最大数) + max demand links (要求リンクの最大数) = max ip interfaces (IP インターフェースの最大数) + max ipv6 interfaces (IPv6 インターフェースの最大数) + max ip & ipv6 interfaces (IP および IPv6 インターフェースの最大数))。マシンを 1 つのサーバーに接続する 1 つのクライアントとしてのみ使用する場合、このフィールドは 1 に設定します。サーバーでは、このフィールドはそのサーバーに同時に接続できる IPv6 クライアントの最大数に設定します。この場合、十分な数の IPv6 インターフェースが定義済みであることを確認してください。

**num\_if\_and\_if6 #**

許容する TCP/IP および IPv6 インターフェースの最大数。この値は 10 進数です。この数値を「max ip interfaces (IP インターフェースの最大数)」および「max ipv6 interfaces (IPv6 インターフェースの最大数)」と併せた値が、サーバー・リンク、クライアント・リンク、および要求リンクの最大数の合計を上回ってはいけません (max server links (サーバー・リンクの最大数) + max client links (クライアント・リンクの最大数) + max demand links (要求リンクの最大数) = max ip interfaces (IP インターフェースの最大数) + max ipv6 interfaces (IPv6 インターフェースの最大数) + max ip & ipv6 interfaces (IP および IPv6 インターフェースの最大数))。マシンを 1 つのサーバーに接続する 1 つのクライアントとしてのみ使用する場合、このフィールドは 1 に設定します。サーバーでは、このフィールドはそのサーバーに同時に接続できる IP および IPv6 クライアントの最大数に設定します。この場合、十分な数の IP および IPv6 インターフェースが定義済みであることを確認してください。

**num\_hdlc #**

アクティブにできる同時非同期 PPP セッション (サーバー、クライアント、および要求) の最大数。このフィールドは 10 進数です。この値は、サーバー・リンク、クライアント・リンク、および要求リンクの最大数の合計を上回ってはいけません ([max server connections (サーバー接続の最大数) + max client connections (クライアント接続の最大数) + max demand connections (要求接続の最大数)] = max async hdlc attachments (非同期 HDLC 接続機構の最大数) = [max ip interfaces (IP インターフェースの最大数) + max ipv6 interfaces (IPv6 インターフェースの最大数) + max ip & ipv6 interfaces (IP および IPv6 インターフェースの最大数)]))。

## オプション・キーワード

これらのキーワードはグローバル・デフォルト LCP オプションを変更します。

**txacm 0xXXXXXXXX**

非同期文字マップを送信します。

**-negacm**

非同期文字マッピングをネゴシエーションしません。このオプションを含むピア構成情報フレームをリジェクトします。

**-negmru**

MRU (最大受信単位) をネゴシエーションしません。このオプションを含むピア構成情報フレームをリジェクトします。

**mru #**

必要な MRU。デフォルトは 1500 です。

<b>-negacf</b>	ACF (アドレス制御フィールド) 圧縮をネゴシエーションしません。ACF は圧縮されません。このオプションを含むピア構成情報フレームをリジェクトします。
<b>-negprotocolcompress</b>	プロトコル圧縮をネゴシエーションしません。通常 PPP プロトコル・フィールドは、ネットワーク・プロトコルごとに 1 バイトずつ圧縮されます。これは、フレームの受信および送信の両方に対するこのオプションのネゴシエーションを使用不可にします。

## /etc/ppp/if\_conf ファイル

このファイルでは、サーバー TCP/IP インターフェースすべてが定義されます。構成ファイルでは、空白行および # (ポンド記号) で始まる行は無視されます。インターフェース定義内で、空白行および # (ポンド記号) で始まる行を使用しないでください。これらの行は、インターフェース定義間のみで使用してください。

### キーワード

<b>interface</b>	新規のインターフェース定義が開始されていることを示します。
<b>ip</b> および <b>ipv6</b>	このインターフェースに使用するプロトコル (1 つまたは複数) を指定し、 <b>local_ip</b> 、 <b>local_ip6</b> 、 <b>remote_ip</b> 、および <b>remote_ip6</b> キーワードと一致するようにします。これらのキーワードは単独でも複数の組み合わせでも使用できます。
<b>server</b>	インターフェースがサーバー接続であることを示します。

以下のキーワードが必要です。

**local\_ip** xxx.yyy.zzz.qqq

**remote\_ip** xxx.yyy.zzz.qqq

**local\_ip6** ::XXXX:XXXX:XXXX:XXXX

**remote\_ip6** ::XXXX:XXXX:XXXX:XXXX

これらのアドレスは対で見たときに異ならなければなりません、ローカル IP および IPv6 アドレスは PPP インターフェースすべてに対して同じにすることができます。指定されたサーバーでは、リモート・アドレスは固有でなければなりません。「interface」「server」エントリーには、「smitty PPP IP Interfaces (smitty PPP IP インターフェース)」メニューを使用してインターフェースを構成すると、**local\_ip** アドレスおよび **remote\_ip** アドレスのみが入れます。**remote\_ip6** と **local\_ip6** は、「smitty PPP IPv6 Interfaces (smitty PPP IPv6 インターフェース)」メニューが使用された場合に、これらのエントリーに入れます。結局、4 つすべてが入れるのは、「smitty PPP IP および IPv6 インターフェース (smitty PPP IP and IPv6 インターフェース)」が使用された場合です。

## キーワード

### client

これは IPv6 オプションのみです。すべての IPv6 接続には、クライアント・インターフェースが必要です。アドレスは、システム型式と ID に基づいてランダムに生成されます。アドレスをゼロで埋める (::0:0:0:0 または単に ::) ことを選択して、サーバーがクライアントに IPv6 アドレスを割り当てるようにすることができます。例として **if\_conf** ファイル・エントリーを以下に示します。

```
interface
client
ipv6
local_ip6 ::0000:0000:0000:0000
```

### demand

```
interface
client
ip
ipv6
local_ip6 ::0007:0000:0000:4445
```

local\_XXX と remote\_XXX があり、プロトコル・タイプ (IP、IPv6、あるいはその両方) によって決まります。認証を行うホスト (サーバー) との接続を確立するために、引用符で囲まれたコマンド・ストリングも必要です。例として **if\_conf** ファイル・エントリーを以下に示します。

```
interface
demand
ipv6
local_ip6 ::0007:0000:0000:4444
remote_ip6 ::0009:0000:0000:5555
dcmd "exec /usr/sbin/pppattachd /dev/tty3 demand ipv6 >/dev/tty3 nodaemon"
```

```
interface
demand
ip
ipv6
local_ip 44.44.44.46
remote_ip 66.66.66.66
netmask 255.255.255.0
local_ip6 ::0007:0000:0000:4444
remote_ip6 ::0009:0000:0000:6666
dcmd "exec /usr/sbin/pppattachd /dev/tty4 demand ip ipv6 >/dev/tty4 nodaemon"
```

## オプション・キーワード

**netmask** xxx.xxx.xxx.xxx

IPv4 インターフェースに対するネットマスクを指定します。

## 終了状況

このコマンドは、以下の終了値を戻します。

0 正常終了。  
>0 エラーが発生しました。

## セキュリティ

アクセス制御: このコマンドを実行するには、root 権限を持っていないければなりません。

## 例

/ect/ppp/lcp\_config ファイルの例:



```
# Comment line
server_name pppclient
lcp_server 0
lcp_client 3
lcp_demand 2
num_if 1
num_if6 2
num_if_and_if6 2
num_hdlc 5
```

#### **/ect/ppp/if\_conf** ファイルの例:

```
# Sample ip server configuration information.
# Note that the complete stanza does not contain
# comments or blank lines
interface
server
ip
local_ip 129.35.130.45
remote_ip 129.35.131.191
netmask 255.255.240.0
```

```
# Sample ipv6 server configuration information.
# Note that the complete stanza does not contain
# comments or blank lines
interface
server
ipv6
local_ip6 ::0009:2313:4C00:3193
remote_ip6 ::0009:2313:4C00:3194
```

```
#However between stanzas one can have blank or
# comment lines.
```

```
interface
server
ip
local_ip 129.35.130.45
remote_ip 129.35.131.196
netmask 255.255.240.0
```

```
interface
server
ip
local_ip 129.35.130.45
remote_ip 129.35.131.197
netmask 255.255.240.0
```

```
interface
server
ip
local_ip 129.35.130.45
remote_ip 129.35.131.201
netmask 255.255.240.0
```

```
interface
server
ip
local_ip 129.35.130.45
remote_ip 129.35.131.212
netmask 255.255.240.0
```

上記の構成ファイルは、以下のように IP および IPv6 インターフェースをインストールするサブシステムとなります。

```

pp0: flags=71<UP,POINTOPOINT,NOTRAILERS>
    inet 129.35.130.45 --> 129.35.131.191 netmask 0xfffff000
pp1: flags=31<UP,POINTOPOINT,NOTRAILERS>
    inet 129.35.130.45 --> 129.35.131.196 netmask 0xfffff000
pp2: flags=31<UP,POINTOPOINT,NOTRAILERS>
    inet 129.35.130.45 --> 129.35.131.197 netmask 0xfffff000
pp3: flags=31<UP,POINTOPOINT,NOTRAILERS>
    inet 129.35.130.45 --> 129.35.131.201 netmask 0xfffff000
pp4: flags=31<UP,POINTOPOINT,NOTRAILERS>
    inet 129.35.130.45 --> 129.35.131.212 netmask 0xfffff000
pp5: flags=30<POINTOPOINT,NOTRAILERS>
    inet netmask

```

注: pp5 は、`/etc/ppp/lcp_config` file (`lcp_client 1`) の `lcp_client` キーワードの結果です。IP および IPv6 クライアント・インターフェースには、サーバーとの接続が確立され、IP が IPCP/IPV6CP によってネゴシエーションされるまで、アドレスが関連付けられません。唯一の例外は、要求クライアント・インターフェースの場合です。これらのインターフェースは独自のアドレスを指定し、それをネゴシエーション時に要求します。当然、これらのインターフェースには、PPP サブシステムが開始されるとすぐにそれぞれの IP および IPv6 アドレスが関連付けられます。

## ファイル

<code>/usr/sbin/pppcontrold</code>	<b>pppcontrold</b> デーモンが入っています。
<code>/etc/ppp/lcp_config</code>	サブシステムを構成します ( <b>lcp_config</b> は SMIT によって生成されなければなりません)。
<code>/etc/ppp/if_conf</code>	TCP/IP インターフェースを構成します ( <b>if_conf</b> は SMIT によって生成されなければなりません)。
<code>/etc/ppp/pppcontrold.pid</code>	<b>pppcontrold</b> プロセス <code>id</code> が入っています。
<code>/etc/ppp/ppp.conf</code>	<b>strload</b> コマンドへの入力が入っています。

## 関連情報

**pppattachd** デーモン、**pppdial** コマンド。

**startsrc** コマンド、**stopsrc** コマンド。

**syslog** サブルーチン。

ネットワークおよびコミュニケーションのマネージ の非同期 PPP サブシステムのセクション。

オペレーティング・システムおよびデバイスのマネージのシステム・リソース・コントローラーのセクションでは、サブシステム、サブサーバー、およびシステム・リソース・コントローラーについて説明します。

---

## pppdial コマンド

### 目的

PPP (Point-to-Point プロトコル) サブシステムが使用できるようにリモート・システムとの非同期接続を確立します。

### 構文

```
pppdial [ -t TimeOut ] [ -v ] [ -d VerboseFile ] -f ChatFile
```

## 説明

**pppdial** コマンドは、非同期デバイスにリモート・システムとの接続を確立する機能を提供します。これは、ダイアログをモデムおよびリモート・システムとともに PPP フレームの送信先に実行するための手段として **pppattachd** デモンとともに使用されます。**pppdial** コマンドは、ダイアログが発生するデバイスとして、標準入力 (stdin) および標準出力 (stdout) を使用します。

エラーおよびメッセージは **syslog** 機能を使用してログに記録されます。

## フラグ

<b>-d</b> <i>VerboseFile</i>	チャット・アクティビティー記録を <i>VerboseFile</i> に記録します。 <i>VerboseFile</i> が存在しない場合は、 <b>pppdial</b> コマンドによって作成されます。 <i>VerboseFile</i> が存在する場合は、 <b>pppdial</b> コマンドは既存ファイルに出力を追加します。
<b>-f</b> <i>ChatFile</i>	tty デバイスに発生するダイアログを含むファイルを指定します。 <i>ChatFile</i> の内容は、基本ネットワーク・ユーティリティー (BNU) および UNIX 間コピー・プログラム (UUCP) の構文に従います。
<b>-t</b> <i>TimeOut</i>	チャット・アクティビティーの <b>Expect</b> フェーズの間にタイムアウトになるまで待機すべき秒数を指定します。
<b>-v</b>	syslog 機能を使用して、チャット・アクティビティーを記録します。

## 終了状況

このコマンドは、以下の終了値を戻します。

0	正常終了。
>0	エラーが発生しました。

## セキュリティ

アクセス制御: 任意のユーザー

## 例

リモート・システムとの接続を確立するには、コマンド行に以下のように 1 行で入力してください。

```
/usr/sbin/pppattachd client ip /dev/tty0 connect "/usr/sbin/pppdial  
-v -f /home/pppuser/dialer.file"
```

/home/pppuser/dialer.file という名前の *ChatFile* には、以下のものが含まれています。

```
''  
atdt4311088  
CONNECT  
\\d\\n  
ogin  
pppuser  
ssword  
pppuserpwd
```

それぞれの意味は以下のとおりです。

```
''          Expect a nul string  
atdt4311088 Send the modem the dial command  
           4311088 is the phone number to dial  
CONNECT    Expect connect from the modem  
\\d\\n      Delay for 1 second then send a new line  
ogin       Expect the string ogin
```

```
pppuser      Send the string pppuser
              pppuser is the user id on the remote system
ssword       Expect the string ssword
pppuserpwd   Send the string pppuserpwd
              pppuserpwd is the password of the user pppuser on the
              remote system
```

リモート・システムには、パスワード `pppuserpwd` で定義されるユーザー `pppuser` および以下を含む **`$HOME/.profile`** がなければなりません。

```
exec pppattachd server ip ipv6 2>/dev/null
```

これは、大変単純化された例です。この例では、PPP サブシステムがクライアントおよびサーバー (またはリモート) の両システムで実行中であることが必要です。この例では、クライアント・システムが `/dev/tty0` で定義されたモデムを持っていることが必要です。*ChatFile* には、ダイヤルする番号 4311088 が含まれています。リモート・システムにはまた、パスワードで定義されるユーザーおよびリモート・システムで PPP 接続を始動する **`.profile`** がなければなりません。デバイス (`/dev/tty0`)、電話番号、ユーザー、ユーザー・パスワードおよび PPP 接続を始動するメカニズムは変化しますが、サーバー・システムの現行値を反映していなければなりません。

## ファイル

`/usr/sbin/pppdial` `pppdial` コマンドが入っています。

## 関連情報

`pppattachd` デーモン、`pppcontrold` デーモン。

`syslog` サブルーチン。

ネットワークおよびコミュニケーションのマネージ の非同期 PPP サブシステムのセクション。

---

## pppstat コマンド

### 目的

PPP (Point-to-Point プロトコル) サブシステムの RAS (信頼性、可用性、保守容易性) 情報を抽出し、表示します。

### 構文

`pppstat`

### 説明

`pppstat` コマンドは、アクティビティ・リンクの特定の特性をモニターする機能を提供します。以下の情報が、アクティビティ・リンクすべてに表示されます。

### LCP 多重化レイヤー

ローカル MRU

このホストに対する最大受信単位の設定値を指定します。これは、リモート・ホストがローカル・ホストに送信することができるパケットの最大長です。

リモート MRU	このリモート・ホストに対する最大受信単位の設定値を指定します。これは、リモート・ホストに送信することができるパケットの最大長です。
ローカルからピア ACCM	リモート・ホストへのパケット送信に使用される ASYNC 文字マップを指定します。
ピアからローカル ACCM	ローカル・ホストへのパケット送信でリモート・ホストが使用する ASYNC 文字マップを指定します。
ローカルからリモートへのプロトコル・フィールド圧縮	リモート・ホストへのパケット送信でプロトコル圧縮が使用されるかどうかを指定します。
リモートからローカルへのプロトコル・フィールド圧縮	リモート・ホストからローカル・ホストへのパケット送信でプロトコル圧縮が使用されるかどうかを指定します。
ローカルからリモートへのアドレス/制御フィールド圧縮	リモート・ホストへのパケット送信でアドレス/制御フィールド圧縮が使用されているかどうかを指定します。
リモートからローカルへのアドレス/制御フィールド圧縮	リモート・ホストからローカル・ホストへのパケット送信でアドレス/制御フィールド圧縮が使用されているかどうかを指定します。

## PPP ネゴシエーション前の LCP 多重化レイヤー

MRU	パケット受信に対する最大受信単位を指定します。これは、このホストがリモート・ホストとネゴシエーションしようとした値です。
ACCM の受信	ネゴシエーションで使用された初期 remote-to-local ASYNC 文字マップを指定します。
ACCM の送信	ネゴシエーションで使用された初期 local-to-remote ASYNC 文字マップを指定します。
マジック・ナンバー フレーム検査サイズ	ネゴシエーションで試されたマジック・ナンバーを指定します。 ホストがネゴシエーションしようとしたフレーム検査シーケンスの長さを指定します。これは 16 ビットに固定されています。

## HDLC フレーム指示レイヤー

正しくないアドレス・フィールド	パケットが正しくないアドレス・フィールドで受信された回数を指定します。
正しくない制御フィールド	パケットが正しくない制御フィールドで受信された回数を指定します。
オーバー・サイズのパケット	最大受信単位の長さを超える長さのパケットが受信された回数を指定します。
正しくないフレーム検査順序列	パケットが正しくないフレーム検査シーケンスで受信された回数を指定します。
正しいオクテットの着信	有効なパケットに受信されたオクテット数を指定します。
正しいオクテットの発信	パケットに正常に送信されたオクテット数を指定します。
正しいパケットの着信	正常に受信されたパケット数を指定します。
正しいパケットの発信	正常に送信されたパケット数を指定します。

出力は **stdout** に送信されます。メッセージは **stderr** に送信されます。

## 終了状況

このコマンドは、以下の終了値を戻します。

0 正常終了。

>0 エラーが発生しました。

## セキュリティ

アクセス制御: 任意のユーザー

イベントの監査: N/A

## ファイル

`/usr/sbin/pppstat` `pppstat` コマンドが入っています。

## 関連情報

コマンド: `pppdial`。

デーモン: `pppcontrold`、`pppattachd`。

ファイル: `profile`。

サブルーチン: `syslog`。

ネットワークおよびコミュニケーションのマネージ の非同期 PPP サブシステムのセクション。

---

## pprof コマンド

### 目的

一定期間にわたって、すべてのカーネル・スレッドの CPU 使用量を報告します。

### 構文

```
pprof { time | -l pprof.flow | -i tracefile | -d } [ -T bytes ] [ -v ] [ -s ] [ -n ] [ -f ] [ -p ] [ -w ] [-r PURR]
```

### 説明

`pprof` コマンドは、`trace` ユーティリティを使用して、あるインターバルで実行しているすべてのカーネルについての報告をします。未加工プロセス情報は、`pprof.flow` に保存され、5 つのレポートが生成されます。`pprof` コマンドは、以前に生成された `Pprof.flow` を取り出してレポートを再生成することもできます。フラグを指定しない場合、すべてのレポートが生成されます。

### レポートのタイプ

#### `pprof.cpu`

実際の CPU 時間によってソートされたすべてのカーネル・レベルのスレッドをリストします。リストの内容: プロセス名、プロセス ID、親プロセス ID、開始および終了時のプロセス状態、スレッド ID、親スレッド ID、実際の CPU 時間、開始時刻、停止時刻、停止 - 開始。

<b>pprof.start</b>	開始時刻によってソートされたすべてのカーネル・スレッドをリストします。リストの内容: プロセス名、プロセス ID、親プロセス ID、開始および終了時のプロセス状態、スレッド ID、親スレッド ID、実際の CPU 時間、開始時刻、停止時刻、停止 - 開始。
<b>pprof.namecpu</b>	各タイプのカーネル・スレッド (すべて同じ名前で行実行可能) に関する情報をリストします。リストの内容: プロセス名、スレッド数、CPU 時間、合計 CPU 時間の %。
<b>pprof.famind</b>	ファミリーごとにグループ化されたすべてのプロセス (祖先が共通しているプロセス) をリストします。子プロセス名は、親プロセスより字下げされます。リストの内容: 開始時刻、停止時刻、実際の CPU 時間、プロセス ID、親プロセス ID、スレッド ID、親スレッド ID、開始および終了時のプロセス状態、レベル、プロセス名。
<b>pprof.famcpu</b>	すべてのファミリー (祖先が共通しているプロセス) に関する情報をリストします。ファミリーのプロセス名とプロセス ID は必ずしも祖先ではありません。リストの内容: 開始時刻、プロセス名、プロセス ID、スレッド数、合計 CPU 時間。

## フラグ

<b>-d</b>	ユーザーが、コマンド行から <b>trcon</b> と <b>trcstop</b> を実行するのを待ちます。
<b>-f</b>	<b>pprof.famcpu</b> レポートと <b>pprof.famind</b> レポートだけを生成するように指定します。
<b>-l pprof.flow</b>	以前に生成された <b>pprof.flow</b> からレポートを生成するように示します。 <b>pprof.namecpu</b> レポートだけを生成するように指定します。
<b>-i tracefile</b>	<b>tracefile</b> からレポートを生成するように示します。このトレースには、135、106、10C、134、139、465、467、00A などのフックが含まれていなければなりません。
<b>-n</b>	<b>pprof.namecpu</b> レポートだけを生成するように指定します。
<b>-p</b>	<b>pprof.cpu</b> レポートだけを生成するように指定します。
<b>-r PURR</b>	パーセントと CPU 時間の計算では、TimeBase ではなく PURR 時間を使用します。経過時間の計算には影響しません。
<b>-s</b>	<b>pprof.start</b> レポートだけを生成するように指定します。
<b>-T</b>	トレース・カーネル・バッファ・サイズをバイト単位で設定します。デフォルトは 32000 です。
<b>-v</b>	冗長モード (エクストラ詳細の表示) を設定します。
<b>-w</b>	<b>pprof.flow</b> だけを生成するように指定します。
<i>time</i>	システムをトレースする秒数を指定します。

注: パフォーマンス分析ツールへの変更に関する最新情報については、 [/usr/lpp/perfagent/README.perfagent.tools](#) を参照してください。

## 関連情報

**trace** コマンド、 **trcrpt** コマンド、 **filemon** コマンド。

**trcon** サブルーチン、 **trcstop** サブルーチン。

---

## pr コマンド

### 目的

ファイルを標準出力に書き出します。



## 構文

```
pr [ +Page ] [ -Column [ -a ] | -m ] [ -d ] [ -F ] [ -r ] [ -t ] [ -e [ Character ] [ Gap ] ]  
[ -h Header ] [ -i [ Character ] [ Gap ] ] [ -l Lines ] [ -n [ Character ] [ Width ] ] [ -o Offset ]  
[ -s [ Character ] ] [ -w Width ] [ -x [ Character ] [ Width ] ] [ -f ] [ -p ] [ File ... | - ]
```

## 説明

**pr** コマンドは指定されたファイルを標準出力に書き出します。*File* パラメーターの代わりに **-** (負符号) パラメーターを指定するか、またはどちらも指定しないと、**pr** コマンドは標準入力を読み取ります。ページ番号、日付、時間、およびファイル名を含む見出しは、出力を数ページに分けます。

指定されなければ、欄は均等幅になり、少なくとも 1 スペースで区切られます。ページ幅よりも長い行は分割されます。標準出力がワークステーションであれば、**pr** コマンドは終了するまでエラー・メッセージを表示しません。

## フラグ

- Column** 欄数を、*Column* 変数で指定された値に設定します。デフォルト値は 1 です。このオプションは **-m** フラグとともに使用しないでください。複数欄出力では、**-e** および **-i** フラグが想定されます。テキスト桁は、ページの長さを超えてはなりません (**-l** フラグを参照してください)。**-Column** フラグを **-t** フラグとともに使用した場合、最小の行数を使って出力を書き込みます。
- +Page** *Page* 変数で指定されたページ番号から表示を始めます。デフォルト値は 1 です。
- a** 複数の欄が左から右へ横方向に充てんされるように、**-Column** フラグの結果を変更します。例えば、2 つの欄がある場合に、第 1 の入力行は欄 1 に、第 2 の入力行は欄 2 に、第 3 の入力行は欄 1 の 2 行目とこのように表示されます。**-a** フラグを指定しなければ、欄は縦方向に作成されます。
- d** 出力をダブルスペースにします。
- e[Character][Gap]** *Gap+1*、*2\*Gap+1*、*3\*Gap+1* などのようにタブを文字位置に拡張します。*Gap* のデフォルト値は 8 です。入力中のタブ文字は、次のタブ設定と並ぶように適切なスペース数に拡張します。*Character* 変数に値 (数値以外の文字) を指定した場合には、その文字が入力タブ文字になります。*Character* 変数のデフォルト値は、ASCII の TAB 文字です。
- F** 用紙送り文字を使用して、新しいページに進みます。(そうでない場合には、**pr** コマンドが改行文字のシーケンスを発行します。) 標準出力がワークステーションであるときには、最初のページを開始する前に一時停止します。このフラグは **-f** フラグと同等です。
- f** 用紙送り文字を使用して、新しいページに進みます。(そうでない場合には、**pr** コマンドが改行文字のシーケンスを発行します。) 標準出力がワークステーションであるときには、最初のページを開始する前に一時停止します。このフラグは **-F** フラグと同等です。
- h Header** 指定されたヘッダー文字列をページ・ヘッダーとして使用します。**-h** フラグを使用しないと、ページ・ヘッダーはデフォルトの、*File* パラメーターによって指定されたファイル名に設定されます。
- i[Character][Gap]** *Gap+1*、*2\*Gap+1*、および *3\*Gap+1* などのように、文字位置にタブを挿入して、可能な限りホワイト・スペースを置き換えます。*Gap* のデフォルト値は 8 です。*Character* 変数に値 (数値以外の文字) を指定した場合には、その文字が出力タブ文字として使用されます。

<b>-l</b> <i>Lines</i>	66 行のデフォルトを取り消し、ページの長さを <i>Lines</i> 変数で指定した行数に再設定します。 <i>Lines</i> 値がヘッダーとトレーラーの奥行き (行数) の合計よりも小さい場合には、ヘッダーおよびトレーラーは抑制されます ( <b>-t</b> フラグが有効であるときのように)。
<b>-m</b>	ファイルをマージします。標準出力は、欄位置の数に基づいて、 <b>pr</b> コマンドが <i>File</i> パラメーターによって指定されたそれぞれのファイルから 1 行を、均等に固定された幅のテキスト欄に横並びに書き込めるようにフォーマットされます。このフラグは <b>-Column</b> フラグとともに使用しないでください。
<b>-n</b> [ <i>Character</i> ][ <i>Width</i> ]	<i>Width</i> 変数によって指定された桁数に基づいて行番号付けを行います。デフォルト値は 5 桁です。デフォルト出力の場合には、行番号は各テキスト欄または <b>-m</b> フラグが設定されている場合には出力の各行の最初の <i>Width</i> +1 桁位置に示されず、 <i>Character</i> 変数を指定すると (数値以外の文字)、その文字が行番号に追加され、行番号とその行の後続の文字とを区切ります。デフォルトの文字セパレーターは、タブ文字です。
<b>-o</b> <i>Offset</i>	<i>Offset</i> 変数によって指定された文字位置数だけ、各行を字下げします。1 行ごとの文字の総数は、幅とオフセットを合計した値です。 <i>Offset</i> のデフォルト値は 0 です。
<b>-p</b>	出力がワークステーションに送られる場合に、各ページを開始する前に一時停止します。 <b>pr</b> コマンドはワークステーションで警告音を発し、ユーザーが <b>Enter</b> キーを押すのを待ちます。
<b>-r</b>	システムがファイルをオープンできない場合、診断メッセージを表示しません。
<b>-s</b> [ <i>Character</i> ]	適切な数のスペースで区切る代わりに、 <i>Character</i> 変数によって指定された単一文字によって欄を区切ります。 <i>Character</i> 変数のデフォルト値は、ASCII の TAB 文字です。
<b>-t</b>	識別ヘッダーの 5 行およびフッターの 5 行を表示しません。ページの終わりまでスペースを空けずに、各ファイルの最終行の後で停止します。
<b>-w</b> <i>Width</i>	行幅を複数のテキスト欄の出力専用の欄の幅の位置に設定します。 <b>-w</b> オプションも <b>-s</b> オプションも指定しない場合は、デフォルトの幅は 72 になります。 <b>-w</b> オプションを指定せずに <b>-s</b> オプションを指定すると、デフォルトの幅は 512 になります。単一の欄の出力では、入力行は切り捨てません。
<b>-x</b> [ <i>Character</i> ][ <i>Width</i> ]	<b>-n</b> フラグと同じ行番号付け機能を提供します。

## 終了状況

このコマンドは、以下の終了値を戻します。

0	すべてのファイルが正常に書き込まれました。
>0	エラーが発生しました。

## 例

- 見出しおよびページ番号を付けてファイルをプリンターに印刷するには、以下のように入力します。

```
pr prog.c | qprt
```

これにより、ページ・ヘッダーは **prog.c** に追加され、**qprt** コマンドに送信されます。見出しは、ファイルの最終修正日、ファイル名、およびページ番号で構成されています。

- タイトルを指定するには、以下のように入力します。

```
pr -h "MAIN PROGRAM" prog.c | qprt
```

これにより、**prog.c** ファイルはファイル名の代わりに **Main Program** というタイトルを付けて印刷されます。修正日とページ番号も引き続き印刷されます。

3. 複数の欄でファイルを印刷するには、以下のように入力します。

```
pr -3 word.lst | qprt
```

これにより、**word.lst** ファイルが縦 3 欄に印刷されます。

4. 複数のファイルを並べて用紙に印刷するには、以下のように入力します。

```
pr -m -h "Members and Visitors" member.lst visitor.lst | qprt
```

これにより、**member.lst** および **visitor.lst** ファイルが、**Members and Visitors** というタイトルで並んで印刷されます。

5. 後で使用できるようにファイルを変更するには、以下のように入力します。

```
pr -t -e prog.c > prog.notab.c
```

これにより、**prog.c** ファイルのタブ文字はスペースに置き換えられ、その結果が **prog.notab.c** ファイルに入力されます。タブ位置は 8 桁ごと (つまり 9, 17, 25, 33, ...) となります。 **-e** フラグは **pr** コマンドにタブ文字を置換するように指示し、 **-t** フラグはページ・ヘッダーを抑制します。

## ファイル

**/usr/bin/pr**                    **pr** コマンドが入っています。  
**/dev/tty\***                    メッセージを延期します。

## 関連情報

**cat** コマンド、 **qprt** コマンド。

オペレーティング・システムおよびデバイスのマネージのファイルのセクションでは、ファイル、ファイル・タイプ、およびファイルの命名方法について説明します。

オペレーティング・システムおよびデバイスのマネージの入出力ダイレクトのセクションでは、オペレーティング・システムの入出力処理方法について説明します。

*AIX 5L* バージョン 5.3 ナショナル・ランゲージ・サポート ガイドおよびリファレンスのナショナル・ランゲージ・サポートの概要に、照合順序、等価クラス、およびロケールが説明されています。

オペレーティング・システムおよびデバイスのマネージのシェルのセクションでは、シェル、さまざまなタイプ、およびコマンドの解釈の仕方にそれらがどのように影響するかについて説明します。

オペレーティング・システムおよびデバイスのマネージのファイルとディレクトリーのアクセス・モードのセクションでは、ファイルおよびディレクトリーへアクセスするためのファイル所有権および権限について紹介しています。

---

## prctmp コマンド

### 目的

セッション・レコード・ファイルを表示します。

## 構文

`/usr/sbin/acct/prctmp` *File...*

## 説明

管理権限を持つユーザーは、**prctmp** コマンドを入力して、**acctcon1** コマンドで作成したセッション・レコード・ファイル (通常は `/var/adm/acct/nite/ctmp` ファイル) を表示できます。セッション・レコード・ファイルは **acctcon2** コマンドによって接続時間合計アカウント・レコードに変換され、日次アカウント・レポートに組み込まれます。

## セキュリティ

アクセス制御: このコマンドは、**adm** グループのメンバーのみに実行 (x) アクセス権限を与えます。

## 例

セッション・レコード・ファイルを表示するには、以下のように入力します。

```
prctmp /var/adm/acct/nite/ctmp
```

このコマンドは、**acctcon1** コマンドで作成したセッション・レコード・ファイルを表示します。

## ファイル

`/usr/sbin/acct`

アカウントिंग・コマンドへのパス。

`/var/adm/acct/nite`

アカウントिंग・データ・ファイルが入っています。

## 関連情報

オペレーティング・システムおよびデバイスのマネージのシステム・アカウントिंगのセクション。

オペレーティング・システムおよびデバイスのマネージのアカウントिंग・サブシステムの設定のセクション。

---

## prdaily コマンド

### 目的

前日のアカウントング情報の ASCII レポートをフォーマットします。

### 構文

```
/usr/sbin/acct/prdaily [ -X ] [ -l ] [ mmdd ] [ -c ]
```

### 説明

**prdaily** コマンドは **runacct** コマンドによってコールされ、前日のアカウントング・データの ASCII レポートをフォーマットします。レポートは `/var/adm/acct/sum/rprt $mmdd$`  ファイル内にあり、 $mmdd$  はレポートの日付を指定します。

## フラグ

- c** コマンド別に例外リソース使用状況を報告します。このフラグは、当日のアカウントिंग・データだけにしか使用できないことがあります。
- l [mddd]** 指定した日付の例外使用状況を、ログイン ID 別に報告します。当日以外の日付を指定するには、*mddd* 変数を使用します。
- X** 最初の 8 文字に切り捨てることをせずに、各ユーザー名の使用可能なすべての文字を処理します。またこのフラグを使用すると、**prdaily** コマンドが **/var/adm/acct/sum** ディレクトリーではなく **/var/adm/acct/sumx** ディレクトリーを使用します。

## セキュリティ

アクセス制御: このコマンドは、**adm** グループのメンバーのみに実行 (x) アクセス権限を与えます。

## ファイル

<b>/usr/sbin/acct</b>	アカウントिंग・コマンドへのパス。
<b>/usr/sbin/acct/ptelus.awk</b>	ログイン ID 別の例外使用状況に対する制限を計算します。これはシェル・プロシージャーです。
<b>/usr/sbin/acct/ptecms.awk</b>	コマンド名別の例外使用状況の制限を計算します。これはシェル・プロシージャーです。
<b>/var/adm/acct/sum</b>	日次アカウント・レコードの累積ディレクトリー。
<b>/var/adm/acct/sumx</b>	長いユーザー名の処理が要求された場合の日次アカウント・レコードのための累積ディレクトリー。

## 関連情報

**acctcms** コマンド、 **acctcom** コマンド、 **acctmerg** コマンド、 **runacct** コマンド。

アカウントिंग・システム、日次および月次のレポートの準備、およびアカウントिंग・ファイルについての詳細は、オペレーティング・システムおよびデバイスのマネージのシステム・アカウントिंगのセクションを参照してください。

オペレーティング・システムおよびデバイスのマネージのアカウントिंग・サブシステムの設定のセクションで、アカウントING・システムの確立のための必要なステップが説明されています。

---

## preparevsd コマンド

### 目的

仮想共用ディスクを使用可能にします。

### 構文

```
preparevsd {-a | vsd_name...}
```

### 説明

**preparevsd** コマンドは、指定した仮想共用ディスクを停止状態から延期状態にします。仮想共用ディスクは使用可能となります。オープン要求、クローズ要求は受け入れられますが、読み取りおよび書き込み要求

は仮想共用ディスクが活動状態になるまで保留されます。読み取りおよび書き込み要求が延期状態である場合は、このコマンドはその要求を延期状態のままにしておきます。

このコマンドは、System Management Interface Tool (SMIT) を使用して実行できます。SMIT を使用するには、以下のように入力します。

```
smit vsd_mgmt
```

さらに、**Prepare a Virtual Shared Disk** オプションを選択します。

通常的环境では、このコマンドを発行すべきではありません。リカバリー可能仮想共用ディスク・サブシステムはこのコマンドを使用して、制御された方法により共用ディスクを管理します。このコマンドを発行する場合、結果は予測不能です。

## フラグ

**-a** 停止状態にあるすべての仮想共用ディスクが使用可能となるように指定します。

## パラメーター

**vsd\_name** 仮想共用ディスクを指定します。仮想共用ディスクが停止状態にない場合は、エラー・メッセージが出されます。

## セキュリティ

このコマンドを実行するには、**root** 権限が必要です。

## 制約事項

このコマンドはピア・ドメイン内のオンラインのノードから発行する必要があります。オンラインのピア・ドメインに移動するには **startprdomain** コマンドを使用します。既存のピア・ドメイン内で、オンラインの特定ノードに移動するには **startprnode** コマンドを使用します。RSCT ピア・ドメインの作成と管理の詳細については、「*RSCT Administration Guide*」を参照してください。

通常的环境では、このコマンドを発行すべきではありません。リカバリー可能仮想共用ディスク・サブシステムはこのコマンドを使用して、制御された方法により共用ディスクを管理します。このコマンドを発行する場合、結果は予測不能です。

## 例

仮想共用ディスク **vsd1vg1n1** を、停止状態から延期状態にするには、次のように入力します。

```
preparevsd vsd1vg1n1
```

## 位置

```
/opt/rsct/vsd/bin/preparevsd
```

## 関連情報

コマンド: **cfgvsd**、**lsvsd**、**resumevsd**、**startvsd**、**stopvsd**、**suspendvsd**、**ucfgvsd**



---

## preprnode コマンド

### 目的

ピア・ドメインに定義するノードを準備します。

### 構文

```
preprnode [-k] [-h] [-TV] node_name1 [node_name2 ... ]
```

```
preprnode -f | -F { file_name | "-" } [-k] [-h] [-TV]
```

### 説明

**preprnode** コマンドは、コマンドを実行するノードにセキュリティーを準備し、ピア・ドメインに定義できるようにします。これにより、ピア・ドメインの操作をこのノードで実行できるようにするものであり、**mkrpdomain** または **addrpnode** コマンドを使用してピア・ドメインに参加するには、その前にこのコマンドを実行しておく必要があります。

ノードに **mkrpdomain** コマンドを実行する前に、新しいピア・ドメインに定義されるそれぞれのノードごとに **preprnode** コマンドを実行する必要があります。この時、パラメーターとして **mkrpdomain** コマンドを実行するノードの名前を使用します。これにより、新しいそれぞれのノードにピア・ドメイン構成を作成するために必要な権限が **mkrpdomain** ノードに与えられ、追加のセキュリティーがセットアップされます。

ノードに **addrpnode** コマンドを実行する前に、追加するそれぞれのノードごとに **preprnode** コマンドを実行する必要があります。この時、すべてのオンライン・ノードの名前をパラメーターとして指定します。これによりオンライン・ノードは、新しいノードに必要な操作を実行する権限が与えられます。

**preprnode** コマンドは以下の機能を実行します。

1. コマンドに指定されたノード名の公開鍵をトラステッド・ホスト・リストに追加することにより、これらのノード名にトラストを確立します。
2. リソース・モニターおよび制御 (RMC) アクセス制御リスト (ACL) ファイルを変更し、このノード上のピア・ドメイン・リソースを、ピア・ドメイン内の他のノードからアクセスできるようにします。これにより、ノードでピア・ドメイン操作を行うことが可能になります。これらのアクセス変更を有効にするため、RMC サブシステムがリフレッシュされます。
3. RMC リモート接続を使用可能にします。

ピア・ドメインに定義されるノードが既に管理ドメイン内にある場合は、公開鍵を交換する必要はありません。**-k** フラグを使用してこのステップを省略できます。

### フラグ

**-f | -F { file\_name | "-" }**

*file\_name* からノード名のリストを読み取ります。ファイルの 1 行ごとに 1 つのノード名が走査されます。# 記号は、その行の残りの部分 (# が 1 桁目にある場合はその行の全体) がコメントであることを示します。

**-f "-"** または **-F "-"** を使用し、入力ファイルとして **STDIN** を指定します。

**-k** コマンドが公開鍵を交換すべきでないことを指定します。

**-h** コマンドの使用ステートメント (使用法) を標準出力に書き込みます。



- T コマンドのトレース・メッセージを標準エラーに書き込みます。これはソフトウェア・サービス担当者のみが使用します。
- V コマンドの詳細メッセージを標準出力に書き込みます。

## パラメーター

`node_name1 [node_name2 ... ]` そこからのピア・ドメイン・コマンドを受け付けることができるノードを指定します。通常、これはピア・ドメインを形成する時に **mkrpdomain** コマンドを実行するノードの名前です。ピア・ドメインに追加する時に、これは現在ピア・ドメイン内でオンラインになっているノードのリストです。ノード名は、IP アドレスまたは DNS ホストのロングまたはショート・バージョンの名前です。ノード名は IP アドレスに解決できるものでなければなりません。

## セキュリティ

**preprnode** コマンドのユーザーは、アクセス制御リスト (ACL) ファイルに書き込み許可が必要です。許可は ACL ファイルに指定します。ACL ファイルおよびその変更方法について詳しくは、「*RSCT: Administration Guide*」を参照してください。

## 終了状況

- 0 コマンドが正常に実行されました。
- 1 RMC でエラーが起きました。
- 2 コマンド行インターフェース・スクリプトでエラーが起きました。
- 3 コマンド行に入力したフラグが誤りです。
- 4 コマンド行に入力したパラメーターが誤りです。
- 5 誤ったコマンド行入力により、エラーが起きました。

## 制約事項

このコマンドは、ピア・ドメインに定義されているノードで実行する必要があります。

## 実装上の固有な条件

このコマンドは、AIX 用の Reliable Scalable Cluster Technology (RSCT) ファイルセットの一部です。

## 標準入力

**-f "-"** または **-F "-"** フラグが指定されているときは、このコマンドで 1 つ以上のノード名が標準入力から読み取られます。

## 標準出力

**-h** フラグが指定されている場合は、このコマンドの使用状況の説明文が標準出力に書き込まれます。詳細メッセージはすべて、標準出力に書き込まれます。

## 標準エラー

トレース・メッセージはすべて、標準エラーに書き込まれます。

## 例

1. **mkrpdomain** を **nodeA** から実行するとします。 **nodeB**、**nodeC**、**nodeD** を新しいピア・ドメインの **ApplDomain** に定義するための準備として、次のコマンドを **nodeB**、**nodeC**、**nodeD** で実行します。

```
preprnode nodeA
```

2. **nodeA** と **nodeB** が **ApplDomain** でオンラインになっているとします。 **nodeC** を既存のドメインに追加する準備をするため、次のコマンドを **nodeC** で実行します。

```
preprnode nodeA nodeB
```

あるいは代わりの方法として、以下の内容を持つファイル (名前は **onlineNodes**) を作成します。

```
nodeA  
nodeB
```

それから、次のコマンドを **nodeC** で実行します。

```
preprnode -f onlineNodes
```

## 位置

**/usr/sbin/rsct/bin/preprnode**

## ファイル

アクセス制御リスト (ACL) ファイル **/var/ct/cfg/ctrmc.acls** が変更されます。このファイルがない場合は、作成されます。

## 関連情報

ブック: 「*RSCT: Administration Guide*」 (ピア・ドメイン操作についての説明)

コマンド: **addrpnode**、**lsrpdomain**、**lsrpnnode**、**mkrpdomain**。

情報ファイル: **rmccli** (RMC 関連のコマンドの一般情報)

ファイル: **ctrmc.acls**

---

## prev コマンド

### 目的

前回のメッセージを表示します。

### 構文

```
prev [ +Folder ] [ -header | -noheader ] [ -showproc CommandString | -noshowproc ]
```

### 説明

**prev** コマンドは、フォルダー内の前回のメッセージを表示します。**prev** コマンドは **prev** 値を指定したときの **show** コマンドに類似しています。

**prev** は、認識しないすべてのフラグを **showproc** プログラムへ渡します。

## フラグ

<b>+Folder</b>	表示したいメッセージが入っているフォルダーを指定します。
<b>-header</b>	表示するメッセージの記述を 1 行で表示します。記述には、フォルダー名およびメッセージ番号が含まれています。このフラグがデフォルトです。
<b>-help</b>	コマンド構文、使用可能なスイッチ (トグル)、およびバージョン情報をリストします。 <b>注:</b> メッセージ・ハンドラー (MH) の場合は、このフラグの名前は完全に つづらなければなりません。
<b>-noheader</b>	各メッセージの 1 行目の記述を表示しないようにします。
<b>-noshowproc</b>	<b>/usr/bin/cat</b> コマンドを使用して、前のコマンドをリストします。
<b>-showproc</b> <i>CommandString</i>	指定されたコマンド・ストリングを使用してリスト作成を実行します。

## プロファイル・エントリー

以下のエントリーは、*UserMhDirectory/.mh\_profile* ファイルの一部です。

Current-Folder:	デフォルトの現行フォルダーを設定します。
Path:	<i>UserMhDirectory</i> を指定します。
showproc:	メッセージの表示に使用されるプログラムを指定します。

## 例

1. 現行フォルダーにある前回のメッセージを表示するには、以下のように入力します。

```
prev
```

システムは次のようなメッセージで応答します。

```
(Message schedule: 10)
```

メッセージのテキストも表示されます。この例では、現行フォルダー `schedule` 内のメッセージ 10 が前回のメッセージです。

2. `meetings` フォルダー内にある前回のメッセージを表示するには、以下のように入力します。

```
prev +meetings
```

システムは次のようなメッセージで応答します。

```
(Message inbox: 5)
```

この例では、`meetings` フォルダー内のメッセージ 5 が前のメッセージです。

## ファイル

<b>\$HOME/.mh_profile</b>	MH ユーザー・プロファイルが入っています。
<b>/usr/bin/prev</b>	<b>prev</b> コマンドが入っています。

## 関連情報

**next** コマンド、**show** コマンド。

**.mh\_alias** ファイル・フォーマット、**.mh\_profile** ファイル・フォーマット。

## printenv コマンド

### 目的

環境変数の値を表示します。

### 構文

**printenv** [ *Name* ]

### 説明

**printenv** コマンドは、環境変数の値を表示します。*Name* パラメーターを指定すると、システムは *Name* パラメーターに関連した値のみを表示します。*Name* パラメーターを指定しない場合には、**printenv** コマンドは現行環境を 1 行に 1 つずつ *Name =Value* の順序で表示します。

環境で定義していない *Name* パラメーターを指定した場合は、**printenv** コマンドは終了状況 1 を返します。それ以外の場合には、0 (ゼロ) の状況を返します。

### 例

1. **MAILMSG** 環境変数の現行値を検索するには、以下のように入力します。

```
printenv MAILMSG
```

2. コマンドは **MAILMSG** 環境変数の値を返します。以下に例を示します。

```
YOU HAVE NEW MAIL
```

### 関連情報

**env** コマンド。

**environment** ファイル。

オペレーティング・システムおよびデバイスのマネージのプロファイルの概要のセクション。

オペレーティング・システムおよびデバイスのマネージのシェルのセクション。

---

## printf コマンド

### 目的

フォーマットされた出力を書き出します。

### 構文

**printf** *Format* [ *Argument...* ]

## 説明

**printf** コマンドは、*Argument* パラメーターを変換してフォーマットし、標準出力へ書き出します。*Argument* パラメーターは *Format* パラメーターの制御下でフォーマットされます。フォーマットされた出力行の長さは、**LINE\_MAX** バイトを超えることはできません。

以下の環境変数が、**printf** コマンドの実行に影響します。

<b>LANG</b>	<b>LC_ALL</b> および対応する環境変数 ( <b>LC_</b> で始まる変数) の両方にロケールが指定されていない場合、ロケール・カテゴリーに使用するロケールを決定します。
<b>LC_ALL</b>	<b>LANG</b> やそのほかの <b>LC_</b> 環境変数の設定で指定されたロケール・カテゴリーの値を上書きするために使用するロケールを決定します。
<b>LC_CTYPE</b>	テキスト・データのバイト・シーケンスを文字として解釈するためのロケールを決定します。例えば、パラメーター中の単一バイト対マルチバイト文字です。
<b>LC_MESSAGES</b>	メッセージを書き込む場合の言語を決定します。
<b>LC_NUMERIC</b>	数値フォーマット設定のロケールを決定します。この環境変数は、 <b>e</b> 、 <b>E</b> 、 <b>f</b> 、 <b>g</b> 、および <b>G</b> 変換文字を使用して書き出される数字のフォーマットに影響します。

*Format* パラメーターは、以下の 3 種類のオブジェクトを含む文字列です。

- 出力ストリームへコピーする非暗号文字。
- 変換指定。各変換指定では 0 またはそれ以上のエントリーが値パラメーター・リストから検索されるようになります。
- 以下のようなエスケープ・シーケンス。出力ストリームへコピーされる時、これらのシーケンスによって、関連するアクションがそのアクションを行えるデバイスに表示されます。

<b>\l</b>	バックスラッシュ
<b>\a</b>	警告
<b>\b</b>	バックスペース
<b>\f</b>	用紙送り
<b>\n</b>	改行
<b>\r</b>	復帰
<b>\t</b>	タブ
<b>\v</b>	垂直タブ
<b>\ddd</b>	<i>ddd</i> は 1 桁、2 桁、または 3 桁の 8 進数。これらのエスケープ・シーケンスは、8 進数で指定した数値を持つバイトとして表示されます。

*Argument* パラメーターは、*Format* パラメーターの制御下で標準出力へ書き出される 1 文字列または複数文字列のリストです。

*Format* パラメーターは、*Argument* パラメーターを満たすのに必要な回数だけ繰り返し使用されます。余分な **c** または **s** 変換指定はどれも、**null** 文字列 *Argument* が提供されたときと同じように評価されます。その他の余分な変換指定は、0 *Argument* が提供されたときと同じように評価されます。*Format* パラメーターに変換指定がなく、*Argument* パラメーターが存在している場合は、その結果は不定になります。

*Format* パラメーターでの各変換指定の構文および指定順序は以下のとおりです。

1. **A %** (パーセント記号)。
2. オプションなし、または 1 つ以上のオプションで、変換指定の意味を変更します。オプションの文字と意味は、以下のとおりです。
  - 変換の結果は、フィールド内に左詰めされます。
  - + 符号付き変換の結果は、常に記号 (+ または -) で始まります。

- blank** 符号付き変換の最初の文字が記号でない場合、結果の前に空白が入ります。空白と + の両方のオプション文字が表示されたときには、空白のオプション文字が無視されます。
- #** このオプションでは、値が代替フォーマットに変換されるように指定します。 **c**、**d**、**i**、**u**、および **s** 変換の場合、このオプションは何の効果も与えません。 **o** 変換の場合、結果の最初の数字が強制的に 0 (ゼロ) になるように精度が高められます。 **x** および **X** 変換の場合、ゼロ以外の値の結果にはそれぞれその前に 0x、または 0X が付きます。 **e**、**E**、**f**、**g**、および **G** 変換の場合、基数文字の後に数値が続かない場合でも、結果には常に基数文字が含まれます。 **g** および **G** 変換の場合、通常の場合と同様に、後続のゼロが結果から除去されることはありません。
- 0** **d**、**i**、**o**、**u**、**x**、**e**、**E**、**f**、**g**、および **G** 変換の場合、先頭にゼロ (符号または基数の指示に続くゼロ) を付けることによってフィールド幅の埋め込みが行われ、スペースでの埋め込みは実行されません。 **0** (ゼロ) および - (負符号) オプションが表示された場合、**0** (ゼロ) オプションは無視されます。 **d**、**i**、**o**、**u**、**x**、および **X** 変換の場合、精度が指定されていれば **0** (ゼロ) オプションは無視されます。

注: その他の変換の場合、動作は不定になります。

3. 最小のフィールド幅を指定するオプションの 10 進数文字列。変換された値の文字数がフィールド幅より少ない場合、フィールド幅で指定された長さになるようにフィールドの左側が埋め込まれます。左調整のオプションを指定している場合には、フィールドの右側が埋め込まれます。変換結果の文字数がフィールド幅より長い場合には、フィールドが拡張されて、変換された結果が入るようにします。切り捨ては行われません。ただし、精度が高い場合、右側の切り捨てが行われることがあります。
4. オプションの精度。この精度は 10 進数の文字列の後の . (ドット) です。精度を指定しない場合には、0 (ゼロ) として扱われます。精度によって以下のものが指定されます。

- **d**、**o**、**i**、**u**、**x**、または **X** 変換について表示される最小桁数。
- **e** および **f** 変換について、基数文字の後に表示される桁数。
- **g** 変換に対する最大有効桁数。
- **s** 変換で文字列から表示する最大バイト数。

5. 適用する変換のタイプを示す文字。以下に例を示します。

- %** 変換を行いません。% (パーセント記号) を表示します。
- d**、**i** 整数値を受け取り、それを符号付き 10 進表記に変換します。精度は、表示する最小桁数を指定します。変換対象の値をそれより少ない桁数で表現できる場合には、値の先頭にゼロが付けられて拡張されます。デフォルトの精度は 1 です。ゼロ値をゼロの精度で変換した結果は、null 文字列です。ゼロを先行文字としてフィールド幅を指定すると、フィールド幅の値になるように先頭にゼロが埋め込まれます。
- o** 整数値を受け取り、それを符号付き 8 進表記に変換します。精度は、表示する最小桁数を指定します。変換対象の値をそれより少ない桁数で表現できる場合には、値の先頭にゼロが付けられて拡張されます。デフォルトの精度は 1 です。ゼロ値をゼロの精度で変換した結果は、null 文字列です。ゼロを先行文字としてフィールド幅を指定すると、フィールド幅の値になるように先頭にゼロが埋め込まれます。フィールド幅の 8 進数値は暗黙指定されません。
- u** 整数値を受け取り、それを符号なし 10 進表記に変換します。精度は、表示する最小桁数を指定します。変換対象の値をそれより少ない桁数で表現できる場合には、値の先頭にゼロが付けられて拡張されます。デフォルトの精度は 1 です。ゼロ値をゼロの精度で変換した結果は、null 文字列です。ゼロを先行文字としてフィールド幅を指定すると、フィールド幅の値になるように先頭にゼロが埋め込まれます。
- x**、**X** 整数値を受け取り、それを 16 進表記に変換します。**x** 変換には文字 abcdef が、**X** 変換には文字 ABCDEF が使用されます。精度は、表示する最小桁数を指定します。精度は、表示する最小桁数を指定します。変換対象の値をそれより少ない桁数で表現できる場合には、値の先頭にゼロが付けられて拡張されます。デフォルトの精度は 1 です。ゼロ値をゼロの精度で変換した結果は、null 文字列です。ゼロを先行文字としてフィールド幅を指定すると、フィールド幅の値になるように先頭にゼロが埋め込まれます。
- f** 浮動小数点値または倍精度値を受け取り、それを [-] ddd.ddd. フォーマットの 10 進表記に変換します。基数文字 (ここでは小数点として示される) の後の桁数は、精度指定と同じです。**LC\_NUMERIC** ロケール・カテゴリーは、このフォーマットで使用する基数文字を決定します。精度を指定しない場合には、6 桁が出力されます。精度が 0 (ゼロ) の場合は、基数文字は表示されません。



- e, E** 浮動小数点値または倍精度値を受け取り、それを指数フォーマット `[-] d.dde{+|-}dd` に変換します。基数文字 (ここでは小数点として示される) の前には 1 桁あり、基数文字の後の桁数は精度指定と同じです。  
**LC\_NUMERIC** ロケール・カテゴリーは、このフォーマットで使用する基数文字を決定します。精度を指定しない場合には、6 桁が出力されます。精度が 0 (ゼロ) の場合は、基数文字は表示されません。**E** 変換文字は、指数の前に e ではなく E を付けた数字を置きます。指数には少なくとも 2 桁が必ず含まれます。ただし、表示される値が 2 桁を超える指数を必要とする場合には、付加指数桁が必要に応じて表示されます。
- g, G** 浮動小数点値または倍精度値を受け取り、有効な桁数を指定する精度でそれを **f** または **e** 変換文字 (**G** 変換の場合には **E**) のスタイルに変換します。後ろのゼロは、結果から取り除かれます。基数文字は、その後ろに数字が続く場合のみ表示されます。使用されるスタイルは、変換される値によって左右されます。スタイル **g** の結果が出されるのは、変換から生じた指数が -4 より小さい場合か、または精度以上である場合に限られます。
- c** 値を文字列として受け取り、その文字列の最初の文字を表示します。
- s** 値を文字列として受け取り、文字列の終わりに達するまでか、または精度によって示された文字数に達するまで、文字列から文字を表示します。精度を指定しない場合には、最初の null 文字までのすべての文字が表示されます。
- b** 値をバックスラッシュ・エスケープ・シーケンスを含む可能性のある文字列として受け取ります。文字列の最後に達するまで、あるいは精度指定によって示されたバイト数に達するまで、変換された文字列からのバイトが表示されます。精度を省略した場合には、最初の null 文字までのすべてのバイトが表示されます。

以下のバックスラッシュ・エスケープ・シーケンスがサポートされます。

- 前述の *Format* パラメータの説明にリストされているエスケープ・シーケンス。各エスケープ・シーケンスは、それが表現する個々の文字に変換されます。
- `\c` (バックスラッシュ c) シーケンス。これは表示されず、それを含む文字列パラメータ内の残りの文字、残りの文字列パラメータ、および *Format* パラメータの付加文字を、**printf** が無視するようにします。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0** 正常終了。  
**>0** エラーが発生しました。

## 例

1. 以下のコマンドを入力します。

```
printf "%5d%4d\n" 1 21 321 4321 54321
```

以下の出力が作成されます。

```
 1 21
3214321
54321 0
```

指定した文字列すべてを表示するために、*Format* パラメータが 3 回使用されています。最後の `%4d` 変換指定を満たすために、**printf** コマンドによって 0 (ゼロ) が出力されます。

2. 以下のコマンドを入力します。

```
printf "%c %c\n" 78 79
```

以下の出力が作成されます。

```
7 7
```



3. 以下の例は、**%%\$** フォーマット指定子をどのように使用すれば引数の順序とは異なる順序で日付を印刷できるかを示します。

```
printf ("%1$s, %3$d, %2$s, %4d:%5$.2d", weekday, month, day, hour, min);  
Sunday, 3. July, 10:02  
(weekday, day. month, hour:min)
```

## ファイル

**/usr/bin/printf** **printf** コマンドが入っています。

## 関連情報

**/usr/bin/echo** コマンド。

「*AIX 5L Version 5.3 Technical Reference: Base Operating System and Extensions*」の **printf** サブルーチン。

「*AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ*」のプログラマーの入出力の処理の概要のセクションで、低レベル、ストリーム、端末、および非同期入出力インターフェースにおいて使用されるファイル、コマンド、およびサブルーチンについて説明します。

*AIX 5L バージョン 5.3 ナショナル・ランゲージ・サポート ガイド*およびリファレンスのナショナル・ランゲージ・サポートの概要のセクション。

---

## proccred コマンド

### 目的

プロセスのクレデンシャル (有効、実、保管のユーザー ID とグループ ID) を印刷します。

### 構文

**proccred** *ProcessID* ...

### 説明

**/proc** ファイルシステムは、プロセスを制御するメカニズムを提供します。また、プロセスとスレッドの現行状態についての情報もアクセスできますが、これはバイナリー形式です。 **proctools** コマンドは、いくつかの使用できる情報に基づいて **ascii** レポートを提供します。

ほとんどのコマンドは、プロセス ID のリストまたは **/proc/ProcessID** 文字列を入力とします。したがって、シェル拡張 **/proc/\*** を使用することにより、システム内のすべてのプロセスを指定できます。

**proctools** コマンドは、指定されたプロセスの **/proc** から情報を収集し、ユーザーに表示します。 **proctools** コマンドは **procrun** および **procstop** と同様に、**/proc** インターフェースを使用してプロセスを開始し、停止します。

コマンドが **/proc** から収集した情報は、プロセスの現行状態のスナップショットであり、したがって停止したプロセス以外は、刻々と変化します。

**proccred** コマンドは、プロセスのクレデンシャル (有効、実、保管のユーザー ID とグループ ID) を印刷します。

## フラグ

*ProcessID*

プロセス ID を指定します。

## 例

1. プロセス 5046 のクレデンシャルを表示するには、次のように入力します。

```
proccred 5046
```

## ファイル

**/proc**

**/proc** ファイルシステムが入っています。

## 関連情報

**procfiles** コマンド、**procflags** コマンド、**procldd** コマンド、**procmmap** コマンド、**procrun** コマンド、**procsig** コマンド、**procstack** コマンド、**proctop** コマンド、**proctree** コマンド、**procpwait** コマンド、**procpwdx** コマンド。

---

## procfiles コマンド

### 目的

プロセスがオープンしたすべてのファイル・ディスクリプターの情報をレポートします。

### 構文

```
procfiles [ -F ] [ -n ] ProcessID ...
```

### 説明

**/proc** ファイルシステムは、プロセスを制御するメカニズムを提供します。また、プロセスとスレッドの現行状態についての情報もアクセスできますが、これはバイナリー形式です。 **proctools** コマンドは、いくつかの使用できる情報に基づいて **ascii** レポートを提供します。

ほとんどのコマンドは、プロセス ID のリストまたは **/proc/ProcessID** 文字列を入力とします。したがって、シェル拡張 **/proc/\*** を使用することにより、システム内のすべてのプロセスを指定できます。

**proctools** コマンドは、指定されたプロセスの **/proc** から情報を収集し、ユーザーに表示します。 **proctools** コマンドは **procrun** および **proctop** と同様に、 **/proc** インターフェースを使用してプロセスを開始し、停止します。

コマンドが **/proc** から収集した情報は、プロセスの現行状態のスナップショットであり、したがって停止したプロセス以外は、刻々と変化します。

通常ファイルには、そのファイルのオープンに使用されたモードに基づく権限があります。通常ファイル以外のどのファイルにも、0 アクセス・モードがあります。

**procfiles** コマンドは、プロセスがオープンしたすべてのファイル・ディスクリプターの情報をレポートします。 **-n** オプションを使用すると、対応するファイル名も表示します。

## フラグ

<b>-F</b>	別のプロセスが制御を持っていたとしても、強制的に <code>procfiles</code> がターゲット・プロセスの制御を得るようにします。
<b>-n</b>	ファイル・ディスクリプターが参照するファイル名を印刷します。
<i>ProcessID</i>	プロセス ID を指定します。

## 例

1. プロセス 11928 がオープンしたファイル・ディスクリプターの状況と制御情報を表示するには、次のように入力します。

```
procfiles 11928
```

このコマンドの出力は以下のような情報です。

```
11928 : -sh
Current rlimit: 2000 file descriptors
0: S_IFCHR mode:0622 dev:10,4 ino:2584 uid:100 gid:100 rdev:28,1
O_RDONLY
1: S_IFCHR mode:0622 dev:10,4 ino:2584 uid:100 gid:100 rdev:28,1
O_RDONLY
2: S_IFCHR mode:0622 dev:10,4 ino:2584 uid:100 gid:100 rdev:28,1
O_RDONLY
63: S_IFREG mode:0600 dev:10,8 ino:311 uid:100 gid:100 rdev:40960,10317
O_RDONLY size:2574
```

2. プロセス 15502 がオープンしたファイル・ディスクリプターの名前、状況、制御情報を表示するには、次のように入力します。

```
procfiles -n 15502
```

このコマンドの出力は以下のような情報です。

```
15502 : /home/guest/test
Current rlimit: 2000 file descriptors
0: S_IFCHR mode:0622 dev:10,4 ino:2584 uid:100 gid:100 rdev:28,1
O_RDONLY
1: S_IFCHR mode:0622 dev:10,4 ino:2584 uid:100 gid:100 rdev:28,1
O_RDONLY
2: S_IFCHR mode:0622 dev:10,4 ino:2584 uid:100 gid:100 rdev:28,1
O_RDONLY
3: S_IFREG mode:0644 dev:10,7 ino:26 uid:100 gid:100 rdev:0,0
O_RDONLY size:0 name:/tmp/foo
```

## ファイル

`/proc` `/proc` ファイルシステムが入っています。

## 関連情報

**proccred** コマンド、**procflags** コマンド、**procldd** コマンド、**procmmap** コマンド、**procrun** コマンド、**procsig** コマンド、**procstack** コマンド、**proctop** コマンド、**proctree** コマンド、**proctwait** コマンド、**procwdx** コマンド。

---

## procflags コマンド

### 目的

指定されたプロセスの各スレッドについて、**/proc** トレース・フラグ、**pending** および **held** シグナル、およびその他の **/proc** 状況情報を印刷します。

### 構文

```
procflags [ -r ] ProcessID ...
```

### 説明

**/proc** ファイルシステムは、プロセスを制御するメカニズムを提供します。また、プロセスとスレッドの現行状態についての情報もアクセスできますが、これはバイナリー形式です。 **proctools** コマンドは、いくつかの使用できる情報に基づいて **ascii** レポートを提供します。

ほとんどのコマンドは、プロセス ID のリストまたは **/proc/ProcessID** 文字列を入力とします。したがって、シェル拡張 **/proc/\*** を使用することにより、システム内のすべてのプロセスを指定できます。

**proctools** コマンドは、指定されたプロセスの **/proc** から情報を収集し、ユーザーに表示します。 **proctools** コマンドは **procrun** および **procstop** と同様に、**/proc** インターフェースを使用してプロセスを開始し、停止します。

コマンドが **/proc** から収集した情報は、プロセスの現行状態のスナップショットであり、したがって停止したプロセス以外は、刻々と変化します。

**procflags** コマンドは、指定されたプロセスの各スレッドについて、**/proc** トレース・フラグ、保留シグナルと保持シグナル、およびその他の **/proc** 状況情報を印刷します。オプション **-r** を使用すると、マシン・レジスターの内容が印刷され、プロセスは該当のイベントで停止されます。該当のイベントは、**<sys/procfs.h>** で定義された、**PR\_REQUESTED**、**PR\_FAULTED**、**PR\_SYSENTRY**、および **PR\_SYSEXIT** です。

### フラグ

<b>-r</b>	プロセスが該当のイベントで停止した場合に、現行マシン・レジスターの状態を表示します。
<i>ProcessID</i>	プロセス ID を指定します。

### 例

1. プロセス 5046 のトレース・フラグを表示するには、次のように入力します。

```
procflags 5046
```

このコマンドの出力は以下のような情報です。

```
5046 : -sh
data model = _ILP32 flags = PR_FORK
/4289: flags = PR_ASLEEP | PR_NOREGS
```

2. 該当のイベントで停止された、プロセス 5040 のトレース・フラグとレジスター値を表示するには、次のように入力します。

```
procflags -r 5040
```

このコマンドの出力は以下のような情報です。

```
5040 : ls
data model = _ILP32 flags = PR_FORK
/6999: flags = PR_STOPPED | PR_ISTOP
why = PR_FAULTED what = FLTBPT what = kfork
gpr0 = 0x0          gpr1 = 0x2ff227b0      gpr2 = 0xf0083bec
gpr3 = 0x2ff22cb3  gpr4 = 0x11          gpr5 = 0x65
gpr6 = 0x50        gpr7 = 0x0          gpr8 = 0x41707a7c
gpr9 = 0x4c4f47    gpr10 = 0x80000000   gpr11 = 0x34e0
gpr12 = 0x0        gpr13 = 0xdeadbeef  gpr14 = 0x1
gpr15 = 0x2ff22c0c gpr16 = 0x2ff22c14  gpr17 = 0x0
gpr18 = 0xdeadbeef gpr19 = 0xdeadbeef  gpr20 = 0xdeadbeef
gpr21 = 0xdeadbeef gpr22 = 0x10        gpr23 = 0xfd
gpr24 = 0x2f       gpr25 = 0x2ff227f0  gpr26 = 0x0
gpr27 = 0x2ff22d87 gpr28 = 0x2ff22cb3  gpr29 = 0x0
gpr30 = 0x0        gpr31 = 0xf0048260  iar = 0xd01be900
msr = 0x2d032     cr = 0x28222442     lr = 0xd01d9de0
ctr = 0xec        xer = 0x0          fpscr = 0x0
fpSCRx = 0x0
```

## ファイル

**/proc** **/proc** ファイルシステムが入っています。

## 関連情報

**proccred** コマンド、**procfiles** コマンド、**procldd** コマンド、**procmmap** コマンド、**procrun** コマンド、**procsig** コマンド、**procstack** コマンド、**proctop** コマンド、**proctree** コマンド、**procpwait** コマンド、**procpwdx** コマンド。

---

## procldd コマンド

### 目的

プロセスによりロードされたオブジェクトをリストします。このオブジェクトには、**dlopen()** を使用して明示的に接続された共有オブジェクトを含みます。

### 構文

```
procldd [ -F ] ProcessID ...
```

### 説明

**/proc** ファイルシステムは、プロセスを制御するメカニズムを提供します。また、プロセスとスレッドの現行状態についての情報もアクセスできますが、これはバイナリー形式です。 **proctools** コマンドは、いくつかの使用できる情報に基づいて **ascii** レポートを提供します。

ほとんどのコマンドは、プロセス ID のリストまたは **/proc/ProcessID** 文字列を入力とします。したがって、シェル拡張 **/proc/\*** を使用することにより、システム内のすべてのプロセスを指定できます。

**proctools** コマンドは、指定されたプロセスの **/proc** から情報を収集し、ユーザーに表示します。 **proctools** コマンドは **procrun** および **proctop** と同様に、 **/proc** インターフェースを使用してプロセスを開始し、停止します。

コマンドが **/proc** から収集した情報は、プロセスの現行状態のスナップショットであり、したがって停止したプロセス以外は、刻々と変化します。

**procldd** コマンドは、プロセスによりロードされたオブジェクトをリストします。このオブジェクトには、**dlopen()** を使用して明示的に接続された共有オブジェクトが含まれます。必要な情報はすべて、**/proc/ProcessID/map** ファイルから収集されます。

## フラグ

**-F** 別のプロセスが制御を持っていたとしても、強制的に **procldd** がターゲット・プロセスの制御を得るようにします。

*ProcessID* プロセス ID を指定します。

## 例

1. プロセス 12644 によりロードされたオブジェクトのリストを表示するには、次のように入力します。

```
procldd 12644
```

このコマンドの出力は以下のような情報です。

```
12644 : -ksh
ksh
/usr/lib/libiconv.a[shr4.o]
/usr/lib/libi18n.a[shr.o]
/usr/lib/nls/loc/en_US
/usr/lib/libcrypt.a[shr.o]
/usr/lib/libc.a[shr.o]
```

## ファイル

**/proc** **/proc** ファイルシステムが入っています。

## 関連情報

**procrcd** コマンド、**procfiles** コマンド、**procflags** コマンド、**procmmap** コマンド、**procrun** コマンド、**procsig** コマンド、**procstack** コマンド、**procstop** コマンド、**proctree** コマンド、**procwait** コマンド、**procwdx** コマンド。

---

## procmmap コマンド

### 目的

プロセスのアドレス・スペース・マップを印刷します。

### 構文

```
procmmap [ -F ] ProcessID ...
```

## 説明

**/proc** ファイルシステムは、プロセスを制御するメカニズムを提供します。また、プロセスとスレッドの現行状態についての情報もアクセスできますが、これはバイナリー形式です。 **proctools** コマンドは、いくつかの使用できる情報に基づいて **ascii** レポートを提供します。

ほとんどのコマンドは、プロセス ID のリストまたは **/proc/ProcessID** 文字列を入力とします。したがって、シェル拡張 **/proc/\*** を使用することにより、システム内のすべてのプロセスを指定できます。

**proctools** コマンドは、指定されたプロセスの **/proc** から情報を収集し、ユーザーに表示します。 **proctools** コマンドは **procrun** および **procstop** と同様に、 **/proc** インターフェースを使用してプロセスを開始し、停止します。

コマンドが **/proc** から収集した情報は、プロセスの現行状態のスナップショットであり、したがって停止したプロセス以外は、刻々と変化します。

**procmmap** コマンドは、プロセスのアドレス・スペース・マップを印刷します。これは、プロセス内のそれぞれのマップされたセグメントの、開始アドレスとサイズを表示します。必要な情報はすべて、 **/proc/ProcessID/mmap** ファイルから収集されます。

## フラグ

**-F** 別のプロセスが制御を持っていたとしても、強制的に **procmmap** がターゲット・プロセスの制御を得るようにします。

*ProcessID* プロセス ID を指定します。

## 例

1. プロセス 12644 のアドレス・スペースを表示するには、次のように入力します。

```
procmmap 12644
```

このコマンドの出力は以下のような情報です。

```
12644 : -ksh
10000000      232K read/exec      ksh
20000ef8      54K  read/write     ksh
d008b100      80K  read/exec      /usr/lib/libiconv.a[shr4.0]
f03e4c70      41K  read/write     /usr/lib/libiconv.a[shr4.0]
d0080100      40K  read/exec      /usr/lib/libi18n.a[shr.o]
f03f0b78      4K   read/write     /usr/lib/libi18n.a[shr.o]
d007a000      11K  read/exec      /usr/lib/nls/loc/en_US
d007d130      8K   read/write     /usr/lib/nls/loc/en_US
d00790f8      2K   read/exec      /usr/lib/libcrypt.a[shr.o]
f03e3508      0K   read/write     /usr/lib/libcrypt.a[shr.o]
d02156c0     2282K read/exec      /usr/lib/libc.a[shr.o]
f03474e0     621K read/write     /usr/lib/libc.a[shr.o]
Total        3380K
```

## ファイル

**/proc** **/proc** ファイルシステムが入っています。



## 関連情報

**proccred** コマンド、**procfiles** コマンド、**procflags** コマンド、**procldd** コマンド、**procrun** コマンド、**procsig** コマンド、**procstack** コマンド、**proctop** コマンド、**proctree** コマンド、**procpwait** コマンド、**procpwdx** コマンド。

---

## procrun コマンド

### 目的

**PR\_REQUESTED** イベントで停止したプロセスを開始します。

### 構文

**procrun** *ProcessID* ...

### 説明

**/proc** ファイルシステムは、プロセスを制御するメカニズムを提供します。また、プロセスとスレッドの現行状態についての情報もアクセスできますが、これはバイナリー形式です。 **proctools** コマンドは、いくつかの使用できる情報に基づいて **ascii** レポートを提供します。

ほとんどのコマンドは、プロセス ID のリストまたは **/proc/ProcessID** 文字列を入力とします。したがって、シェル拡張 **/proc/\*** を使用することにより、システム内のすべてのプロセスを指定できます。

**proctools** コマンドは、指定されたプロセスの **/proc** から情報を収集し、ユーザーに表示します。 **proctools** コマンドは **procrun** および **proctop** と同様に、 **/proc** インターフェースを使用してプロセスを開始し、停止します。

コマンドが **/proc** から収集した情報は、プロセスの現行状態のスナップショットであり、したがって停止したプロセス以外は、刻々と変化します。

**procrun** コマンドは、**PR\_REQUESTED** イベントで停止したプロセスを開始します。

### フラグ

*ProcessID*

プロセス ID を指定します。

### 例

1. **PR\_REQUESTED** イベントで停止したプロセス 30192 を再始動するには、次のように入力します。

```
procrun 30192
```

### ファイル

**/proc**

**/proc** ファイルシステムが入っています。

## 関連情報

**proccred** コマンド、**procfiles** コマンド、**procflags** コマンド、**procldd** コマンド、**procmmap** コマンド、**procsig** コマンド、**procstack** コマンド、**procstop** コマンド、**proctree** コマンド、**procwait** コマンド、**procwdx** コマンド。

---

## procsig コマンド

### 目的

プロセスで定義されたシグナル・アクションをリストします。

### 構文

**procsig** *ProcessID* ...

### 説明

**/proc** ファイルシステムは、プロセスを制御するメカニズムを提供します。また、プロセスとスレッドの現行状態についての情報もアクセスできますが、これはバイナリー形式です。 **proctools** コマンドは、いくつかの使用できる情報に基づいて **ascii** レポートを提供します。

ほとんどのコマンドは、プロセス ID のリストまたは **/proc/ProcessID** 文字列を入力とします。したがって、シェル拡張 **/proc/\*** を使用することにより、システム内のすべてのプロセスを指定できます。

**proctools** コマンドは、指定されたプロセスの **/proc** から情報を収集し、ユーザーに表示します。 **proctools** コマンドは **procrun** および **procstop** と同様に、**/proc** インターフェースを使用してプロセスを開始し、停止します。

コマンドが **/proc** から収集した情報は、プロセスの現行状態のスナップショットであり、したがって停止したプロセス以外は、刻々と変化します。

**procsig** コマンドは、プロセスで定義されたシグナル・アクションをリストします。

### フラグ

*ProcessID*

プロセス ID を指定します。

### 例

1. プロセス 11928 に定義されたすべてのシグナル・アクションをリストするには、次のように入力します。

```
procsig 11928
```

このコマンドの出力は以下のような情報です。

```
HUP      caught
INT      caught
QUIT     caught
ILL      caught
TRAP     caught
ABRT     caught
EMT      caught
FPE      caught
```

KILL	default	RESTART
BUS	caught	
SEGV	default	
SYS	caught	
PIPE	caught	
ALRM	caught	
TERM	ignored	
URG	default	
STOP	default	
TSTP	ignored	
CONT	default	
CHLD	default	
TTIN	ignored	
TTOU	ignored	
IO	default	
XCPU	default	
XFSZ	ignored	
MSG	default	
WINCH	default	
PWR	default	
USR1	caught	
USR2	caught	
PROF	default	
DANGER	default	
VTALRM	default	
MIGRATE	default	
PRE	default	
VIRT	default	
ALRM1	default	
WAITING	default	
CPUFAIL	default	
KAP	default	
RETRACT	default	
SOUND	default	
SAK	default	

## ファイル

`/proc` `/proc` ファイルシステムが入っています。

## 関連情報

`proccred` コマンド、`procfiles` コマンド、`procflags` コマンド、`procldd` コマンド、`procmmap` コマンド、`procrun` コマンド、`procstack` コマンド、`procstop` コマンド、`proctree` コマンド、`procwait` コマンド、`procwdx` コマンド。

---

## procstack コマンド

### 目的

プロセス内のすべてのスレッドについて、16 進アドレスとシンボル名を印刷します。

### 構文

`procstack [ -F ] ProcessID ...`

## 説明

**/proc** ファイルシステムは、プロセスを制御するメカニズムを提供します。また、プロセスとスレッドの現行状態についての情報もアクセスできますが、これはバイナリー形式です。 **proctools** コマンドは、いくつかの使用できる情報に基づいて **ascii** レポートを提供します。

ほとんどのコマンドは、プロセス ID のリストまたは **/proc/ProcessID** 文字列を入力とします。したがって、シェル拡張 **/proc/\*** を使用することにより、システム内のすべてのプロセスを指定できます。

**proctools** コマンドは、指定されたプロセスの **/proc** から情報を収集し、ユーザーに表示します。 **proctools** コマンドは **procrun** および **procstop** と同様に、 **/proc** インターフェースを使用してプロセスを開始し、停止します。

コマンドが **/proc** から収集した情報は、プロセスの現行状態のスナップショットであり、したがって停止したプロセス以外は、刻々と変化します。

**procstack** コマンドは、プロセス内のすべてのスレッドについて 16 進アドレスとシンボル名を印刷します。

## フラグ

**-F**

別のプロセスが制御を持っていたとしても、強制的に **procstack** がターゲット・プロセスの制御を得るようにします。

*ProcessID*

プロセス ID を指定します。

## 例

1. プロセス 11928 の現行スタックを表示するには、次のように入力します。

```
procstack 11928
```

このコマンドの出力は以下のような情報です。

```
11928 : -sh
d01d15c4 waitpid  (? , ? , ?) + e0
10007a1c job_wait  (?) + 144
10020298 xec_switch  (? , ? , ? , ? , ?) + 9c0
10021db4 sh_exec  (? , ? , ?) + 304
10001370 exfile  () + 628
10000300 main  (? , ?) + a1c
10000100 __start  () + 8c
```

2. アプリケーション *appl* のマルチスレッド・プロセス 28243 のすべてのスレッドの現行スタックを表示するには、次のように入力します。

```
procstack 28243
```

このコマンドの出力は次のようなものです。

```
28243 : appl
----- tid# 54321 -----
d0059eb4 _p_nsleep  (? , ?) + 10
d01f1fc8 nsleep  (? , ?) + b4
d026a6c0 sleep  (?) + 34
100003a8 main  () + 98
10000128 __start  () + 8c
----- tid# 43523 -----
d0059eb4 _p_nsleep  (? , ?) + 10
d01f1fc8 nsleep  (? , ?) + b4
```

```
d026a6c0 sleep (?) + 34
10000480 PrintHello (d) + 30
d004b314 _pthread_body (?) + ec
----- tid# 36352 -----
d0059eb4 _p_nsleep (?, ?) + 10
d01f1fc8 nslsleep (?, ?) + b4
d026a6c0 sleep (?) + 34
10000480 PrintHello (c) + 30
d004b314 _pthread_body (?) + ec
```

## ファイル

**/proc** **/proc** ファイルシステムが入っています。

## 関連情報

**proccred** コマンド、**procfiles** コマンド、**proclags** コマンド、**procldd** コマンド、**procmmap** コマンド、**procrun** コマンド、**procsig** コマンド、**procstop** コマンド、**proctree** コマンド、**procwait** コマンド、**procwdx** コマンド。

---

## procstop コマンド

### 目的

**PR\_REQUESTED** イベントでプロセスを停止します。

### 構文

**procstop** *ProcessID* ...

### 説明

**/proc** ファイルシステムは、プロセスを制御するメカニズムを提供します。また、プロセスとスレッドの現行状態についての情報もアクセスできますが、これはバイナリー形式です。 **proctools** コマンドは、いくつかの使用できる情報に基づいて **ascii** レポートを提供します。

ほとんどのコマンドは、プロセス ID のリストまたは **/proc/ProcessID** 文字列を入力とします。したがって、シェル拡張 **/proc/\*** を使用することにより、システム内のすべてのプロセスを指定できます。

**proctools** コマンドは、指定されたプロセスの **/proc** から情報を収集し、ユーザーに表示します。 **proctools** コマンドは **procrun** および **procstop** と同様に、 **/proc** インターフェースを使用してプロセスを開始し、停止します。

コマンドが **/proc** から収集した情報は、プロセスの現行状態のスナップショットであり、したがって停止したプロセス以外は、刻々と変化します。

**procstop** コマンドは、**PR\_REQUESTED** イベントでプロセスを停止します。

### フラグ

*ProcessID* プロセス ID を指定します。

## 例

1. **PR\_REQUESTED** イベントでプロセス 7500 を停止するには、次のように入力します。

```
procstop 7500
```

## ファイル

**/proc** **/proc** ファイルシステムが入っています。

## 関連情報

**proccred** コマンド、**procfiles** コマンド、**proclags** コマンド、**procldd** コマンド、**procmmap** コマンド、**procrun** コマンド、**procsig** コマンド、**procstack** コマンド、**proctree** コマンド、**procwait** コマンド、**procwdx** コマンド。

---

## proctree コマンド

### 目的

指定されたプロセス ID またはユーザーを含むプロセス・ツリーを印刷します。

### 構文

```
proctree [ -a ] [ { ProcessID | User } ]
```

### 説明

**/proc** ファイルシステムは、プロセスを制御するメカニズムを提供します。また、プロセスとスレッドの現行状態についての情報もアクセスできますが、これはバイナリー形式です。 **proctools** コマンドは、いくつかの使用できる情報に基づいて **ascii** レポートを提供します。

ほとんどのコマンドは、プロセス ID のリストまたは **/proc/ProcessID** 文字列を入力とします。したがって、シェル拡張 **/proc/\*** を使用することにより、システム内のすべてのプロセスを指定できます。

**proctools** コマンドは、指定されたプロセスの **/proc** から情報を収集し、ユーザーに表示します。 **proctools** コマンドは **procrun** および **procstop** と同様に、 **/proc** インターフェースを使用してプロセスを開始し、停止します。

コマンドが **/proc** から収集した情報は、プロセスの現行状態のスナップショットであり、したがって停止したプロセス以外は、刻々と変化します。

**proctree** コマンドは、指定されたプロセス ID またはユーザーを含むプロセス・ツリーを印刷します。子プロセスは、その親プロセスの下に字下げされます。すべて数字の引数はプロセス ID と見なされ、それ以外はユーザー・ログイン名と見なされます。デフォルトのアクションは、プロセス 0 の子以外のすべてのプロセスをレポートします。

### フラグ

**-a**

プロセス 0 の子を表示に含めます。デフォルトはこれを除いて表示します。

*ProcessID*

プロセス ID を指定します。

## 例

1. プロセス 12312 の祖先とすべての子を表示するには、次のように入力します。

```
proctree 12312
```

このコマンドの出力は以下のような情報です。

```
4954  /usr/sbin/srcmstr
  7224  /usr/sbin/inetd
    5958  telnetd -a
      13212  -sh
        14718  ./proctree 13212
```

2. プロセス 12312 の祖先と子を、プロセス 0 の子を含めて表示するには、次のように入力します。

```
proctree -a 12312
```

このコマンドの出力は以下のような情報です。

```
1  /etc/init
  4954  /usr/sbin/srcmstr
    7224  /usr/sbin/inetd
      5958  telnetd -a
        13212  -sh
          14724  ./proctree -a 13212
```

## ファイル

**/proc**

**/proc** ファイルシステムが入っています。

## 関連情報

**proccred** コマンド、**procfiles** コマンド、**proclags** コマンド、**procldd** コマンド、**procmmap** コマンド、**procrun** コマンド、**procsig** コマンド、**procstack** コマンド、**procstop** コマンド、**procpwait** コマンド、**procpwdx** コマンド。

---

## procpwait コマンド

### 目的

指定したすべてのプロセスが終了するのを待ちます。

### 構文

```
procpwait [ -v ] ProcessID ...
```

### 説明

**/proc** ファイルシステムは、プロセスを制御するメカニズムを提供します。また、プロセスとスレッドの現行状態についての情報もアクセスできますが、これはバイナリー形式です。 **procpools** コマンドは、いくつかの使用できる情報に基づいて **ascii** レポートを提供します。

ほとんどのコマンドは、プロセス ID のリストまたは **/proc/ProcessID** 文字列を入力とします。したがって、シェル拡張 **/proc/\*** を使用することにより、システム内のすべてのプロセスを指定できます。



proctools コマンドは、指定されたプロセスの **/proc** から情報を収集し、ユーザーに表示します。proctools コマンドは **procrun** および **procstop** と同様に、**/proc** インターフェースを使用してプロセスを開始し、停止します。

コマンドが **/proc** から収集した情報は、プロセスの現行状態のスナップショットであり、したがって停止したプロセス以外は、刻々と変化します。

**procpwait** コマンドは、指定したすべてのプロセスが終了するのを待ちます。

## フラグ

**-v** *ProcessID* 詳細出力を指定します。レポートの宛先は標準出力です。  
プロセス ID を指定します。

## 例

1. プロセス 12942 が終了するのを待ち、状況を表示するには、次のように入力します。

```
procpwait -v 12942
```

このコマンドの出力は以下のような情報です。

```
12942 : terminated, exit status 0
```

## ファイル

**/proc** **/proc** ファイルシステムが入っています。

## 関連情報

**proccred** コマンド、**procfiles** コマンド、**procflags** コマンド、**procldd** コマンド、**procmmap** コマンド、**procrun** コマンド、**procsig** コマンド、**procstack** コマンド、**procstop** コマンド、**proctree** コマンド、**procpwdx** コマンド。

---

## procpwdx コマンド

### 目的

プロセスの現在の作業ディレクトリーを印刷します。

### 構文

```
procpwdx [ -F ] ProcessID ...
```

### 説明

**/proc** ファイルシステムは、プロセスを制御するメカニズムを提供します。また、プロセスとスレッドの現行状態についての情報もアクセスできますが、これはバイナリー形式です。proctools コマンドは、いくつかの使用できる情報に基づいて **ascii** レポートを提供します。

ほとんどのコマンドは、プロセス ID のリストまたは **/proc/*ProcessID*** 文字列を入力とします。したがって、シェル拡張 **/proc/\*** を使用することにより、システム内のすべてのプロセスを指定できます。

proctools コマンドは、指定されたプロセスの **/proc** から情報を収集し、ユーザーに表示します。 proctools コマンドは **procrun** および **procstop** と同様に、 **/proc** インターフェースを使用してプロセスを開始し、停止します。

コマンドが **/proc** から収集した情報は、プロセスの現行状態のスナップショットであり、したがって停止したプロセス以外は、刻々と変化します。

**procwdx** コマンドは、プロセスの現在の作業ディレクトリーを印刷します。

## フラグ

**-F** 別のプロセスが制御を持っていたとしても、強制的に **procfiles** がターゲット・プロセスの制御を得るようにします。

*ProcessID* プロセス ID を指定します。

## 例

1. プロセス 11928 の現在の作業ディレクトリーを表示するには、次のように入力します。

```
procwdx 11928
```

このコマンドの出力は以下のような情報です。

```
11928 : /home/guest
```

## ファイル

**/proc** **/proc** ファイルシステムが入っています。

## 関連情報

**proccred** コマンド、 **procfiles** コマンド、 **procflags** コマンド、 **procldd** コマンド、 **procmmap** コマンド、 **procrun** コマンド、 **procsig** コマンド、 **procstack** コマンド、 **procstop** コマンド、 **proctree** コマンド、 **procwait** コマンド。

---

## prof コマンド

### 目的

オブジェクト・ファイルのプロファイル・データを表示します。

### 構文

```
prof [ -t | -c | -a | -n ] [ -o | -x ] [ -g ] [ -z ] [ -h ] [ -s ] [ -S ] [ -v ] [ -L  
PathName ] [ Program ] [ -m MonitorData ... ]
```

### 説明

**prof** コマンドは、オブジェクト・ファイル *Program* (デフォルトでは **a.out**) に関して **monitor** サブルーチンが収集したプロファイル・データの解釈をします。これは、オブジェクト・ファイル *Program* 内の記号テーブルを読み取って、プロファイル・ファイル (デフォルトでは **mon.out**) と関連付けます。 **prof** コ

マンドは、外部テキストの記号ごとに、その記号のアドレスから次の記号のアドレスまでに費やされた実行時間のパーセンテージ、関数のコール回数、およびコール 1 回当たりの平均ミリ秒数を表示します。

**注:** C++ オブジェクト・ファイルからのシンボルは、使用する前に名前がデマングルされます。

関数のコール回数を記録するには、**-p** フラグを指定した **cc** コマンドを使用して、ファイルをコンパイルしておく必要があります。**-p** フラグによって、コンパイラーは **mcount** サブルーチンのコールを、プログラムの再コンパイル後の各関数に対して生成されたオブジェクト・コードに挿入させます。プログラムの実行中、親関数が子関数をコールするたびに、子関数は **mcount** サブルーチンをコールして、その親子の対に対して専用のカウンターを増やします。**-p** フラグで再コンパイルされていないプログラムには **mcount** サブルーチンが挿入されていないので、どの関数とそのプログラムをコールしたかは記録されません。

また **-p** フラグは、プログラム開始時と終了時に **monitor** サブルーチンをコールする特別プロファイル始動関数がオブジェクト・ファイルに含まれるようにします。プログラムの終了時に **monitor** サブルーチンをコールすると、実際に **mon.out** ファイルを書き込みます。したがって、明示的にメインプログラムを終了させるかまたはメインプログラムから戻るプログラムのみが、**mon.out** ファイルを作成します。

**注:** 生成される出力ファイルの名前を変更するには、**PROF** 環境変数を使用してこれを次のように設定します。

```
PROF=filename:<filename>
```

例えば **PROF=myprof** と設定すると、生成されるファイルは **myprof.out** という名前になります。

ロードされたオブジェクトの位置および名前は **mon.out** ファイルに保管されます。フラグを選択しない場合は、**prof** はこれらの名前を使用します。ほかのオブジェクトにアクセスするには、プログラムを指定するかまたは **-L** オプションを使用しなければなりません。

**注:** 共用ライブラリー・ルーチン呼び出しなど、インポートされた外部ルーチンを呼び出すと、ローカル **glink** コードに対して中間呼び出しが行われ、実ルーチンへの呼び出しがセットアップされます。このコードの実行中にタイマー・クロックがオフになると、時間は **routine.gi** と呼ばれるルーチンにカウントされます。ここで、**routine** はコール先のルーチンです。例えば、**glink** コードが **printf** サブルーチンをコールしている間にタイマーがオフになると、時間は **printf.gi** ルーチンにカウントされます。

## フラグ

相互に排他的なフラグ **a**、**c**、**n**、および **t** は **prof** コマンドによる出力行のソート方法を決定します。

- a** 記号アドレスの昇順でソートします。
- c** コール回数の降順でソートします。
- n** シンボル名の字句別にソートします。
- t** 合計時間のパーセンテージの降順でソートします (デフォルト)。

**注:** フラグ **a**、**c**、**n**、および **t** のうち複数のフラグを同じコマンドで使用しても、**prof** コマンドを正常に実行できます。**prof** コマンドは、これらのフラグのうちコマンド行で最初に見つかったフラグを受け取り、それ以外は無視します。

相互に排他的なフラグ **o** および **x** は、モニターされる各記号のアドレスの表示方法を指定します。

- o** 各アドレスをシンボル名とともに 8 進数で表示します。
- x** 各アドレスをシンボル名とともに 16 進数で表示します。

注: **-o** フラグおよび **-x** フラグの両方を同じコマンドで使用しても、**prof** コマンドを正常に実行できます。**prof** コマンドは、この 2 つのフラグのうちコマンド行で最初に見つかったフラグを受け取り、別のフラグは無視します。

以下のフラグは任意に組み合わせて使用できます。

<b>-g</b>	非グローバル記号 (静的関数) を含めます。
<b>-h</b>	通常はレポートに表示される見出しを抑制します。これは、レポートをさらに処理する場合に有効です。
<b>-L PathName</b>	共用オブジェクトの位置を決めるために代替パス名を使用します。
<b>-m MonitorData</b>	<b>mon.out</b> のかわりに <i>MonitorData</i> からプロファイル・データを取り出します。
<b>-s</b>	<b>mon.sum</b> に要約ファイルを生成します。これは、複数のプロファイル・ファイルを指定するときに有効です。
<b>-S</b>	標準エラーにモニター・パラメーターおよび統計情報の要約を表示します。
<b>-v</b>	すべての表示を抑制し、プロット・フィルターで表示できるように標準出力に図形バージョンのプロファイルを送信します。プロット時に、番号の下限および上限 (デフォルトでは 0 および 100) を指定すると、プロファイルの選択されたパーセンテージをそれに対応する高解像度でプロットすることができます。
<b>-z</b>	0 (ゼロ) コールおよび 0 (ゼロ) 時間に関連している場合でも、すべての記号をプロファイルの範囲内に含めます。

## 例

1. 各記号アドレスでの消費時間を時間でソートして、ヘッダーを付けずに表示するには、以下のように入力します。

```
prof -t -h
```

2. 次の例では、ディレクトリー **/home/score/lib** 内でファイル **runfile** を作成するために使用される共用ライブラリーのローカル・バージョンを取り出します。使用されるデータ・ファイルは **mon.out** ではなく **runfile.mon** です。

```
prof -x -L/home/score/lib runfile -m runfile.mon
```

## ファイル

<b>mon.out</b>	デフォルトのプロファイル。
<b>a.out</b>	デフォルトのオブジェクト・ファイル。
<b>mon.sum</b>	要約プロファイル。

## 関連情報

**cc** コマンド、**nm** コマンド、**gprof** コマンド。

**exit** サブルーチン、**profil** サブルーチン、**monitor** サブルーチン。

オペレーティング・システムおよびデバイスのマネージのコマンドのセクション。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」のサブルーチンの概要のセクション。

---

## proff コマンド

### 目的

パーソナル・プリンター・データ・ストリームを扱うプリンターにテキストをフォーマットします。

### 構文

```
proff [ -LList ] [ -PPrinter ] [ -t ] [ nroffFlags ] [ File ... ]
```

### 説明

**proff** コマンドは、指定されたファイルで **nroff** コマンドを使用して、Quietwriter III プリンター、Quickwriter プリンター、Proprinter プリンターなど、ppds (パーソナル・プリンター・データ・ストリーム) をサポートするプリンター用にテキストをフォーマットします。

ファイルを指定しないと、標準入力を読み取られます。パラメーター値 - (マイナス) は、標準入力を指定します。

### パラメーター

*nroffFlags*            **proff** コマンドによって使用される **nroff** コマンドのフラグを指定し、ppds サポートのプリンター出力用にテキスト・ファイルをフォーマットします。

*File*                    **proff** コマンドが ppds をサポートするプリンター用にフォーマット設定するテキスト・ファイルを指定します。

### フラグ

**-LList**                指定されたリストを **qprt** コマンド用のフラグとして渡します。

単一のフラグを **qprt** コマンドへ渡すには、**-L** フラグとそのすぐ後に渡されている **nroff** コマンドを続けて使用します。以下に例を示します。

**-L-h.**

複数のフラグおよび文字列を **lpr** コマンドへ渡すには、以下のように **-L** フラグとそのすぐ後に " (二重引用符) で囲んだフラグまたは文字列を続けて使用します。

**-L"-h -r -m".**

**-P Printer**            出力を **/etc/qconfig** ファイルのエントリーに対応している指定プリンターへ送信します。デフォルトは、**PRINTER** 環境変数がある場合にはそれから取り込みます。そうでない場合には、システムのデフォルト・キュー名が使用されます。

**-t**                      標準出力へ出力を送信します。

**-**                        標準入力をフォーマット設定プロセスのソースとして使用するよう指定します。

ほかのフラグはすべて **nroff** コマンドに渡されます。

### 例

IBM Proprinter プリンター用に出力を処理する典型的なコマンド・シーケンスを次に示します。

```
proff -t testfile
```

## 環境変数

**PRINTER**      使用したい印刷キューを指定します。

## ファイル

**/usr/share/lib/nterm/tab.ppds**

パーソナル・プリンターのデータ・ストリームを持つプリンター用のドライビング・テーブルが入っています。

**/etc/qconfig**

キューおよびデバイスを記述します。

## 関連情報

**col** コマンド、 **eqn** コマンド、 **lpr** コマンド、 **nroff** コマンド、 **qprt** コマンド、 **tbl** コマンド。

---

## projectl コマンド

### 目的

プロジェクト・ベースの拡張アカウントティング・アクティビティーをサポートします。

### 構文

**projectl add** *projname projnumber* [*comment*] [ { **-d** *projpath* | **-p** [*DN*] } ]

**projectl merge** *sourceprojpath* [ **-d** *targetprojfile* ]

**projectl rm** *projname* [ { **-d** *projpath* | **-p** [*DN*] } ]

**projectl chg** *projname* [ **-p** *pid* [, *pid*] ] [**-f**]

**projectl exec** *projname* <*cmd line*> [**-f**]

**projectl chattr agg** *projname* {**-s**|**-u**} [ { **-d** *projpath* | **-p** [*DN*] } ]

**projectl qpolicy** [ **-g** [*DN*] ]

**projectl qprojs** [**-n**]

**projectl qproj** [*projectname*]

**projectl qapp** *appname*

**projectl** {**chkusr** | **chkgrp** | **chkprojs** | {{**chkadm** | **chkall**}} [**-d** *admpath*]]

**projectl ldusr** [ **-r** ] [ **-a** ]

**projectl unldusr** [ **-a** ]

**projectl ldgrp** [ **-r** ] [ **-a** ]

**projectl unldgrp** [ **-a** ]

**projectl ldprojs -g** [ **-r** ] [ **-a** ]

```

projectl ldprojs -g [DN] -d projpath
projectl ldprojs -p [DN] -d projpath
projectl unldprojs -g [DN] [ -f ] [ -a ]
projectl unldprojs -p [DN]
projectl ldadm -g [name] [ -r ] [ -a ]
projectl ldadm -g [name:]DN | name ] -d admpath
projectl ldadm -p [ [name:]DN | name ] -d admpath
projectl unldadm -g [ -a ]
projectl unldadm -p [ [name:]DN | name ]
projectl ld [ -r ]
projectl ldall [ -d admpath ] [ -r ] [ -a ]
projectl unldall [ -f ] [ -a ]

```

## 説明

**projectl** コマンドのさまざまなサブコマンドは、新規プロジェクトの追加、新規プロジェクトの除去、および特定のアカウントिंग・ポリシーのロードなどの、プロジェクト・ベースの拡張アカウントिंग・アクティビティーを実行します。 **projectl** コマンドのこれらのさまざまなオプションについて、以下に説明します。

## フラグ

- a** システム・リブート時に自動的にポリシーをロードします。
- d** 通常は、参照するプロジェクト定義ファイルまたは管理ポリシー・ファイルがある場所からのパスを指定します。 **merge** サブコマンドの中で使用される場合は、このフラグは、マージされたプロジェクト定義が保管されるターゲット・プロジェクト定義ファイルを指定するためのものです。
- f** **chg** および **exec** サブコマンドと一緒に指定されると、ポリシー・ルールをオーバーライドします。  
**unldall** サブコマンドと一緒に呼び出されると、プロセスに割り当てられたプロジェクトをクリアします。  
**unldprojs** サブコマンドと一緒に呼び出されると、すべてのプロジェクト定義のアンロードを強制します。
- g** プロジェクトおよびポリシーが LDAP リポジトリからダウンロードされることを指定します。
- n** 名前を基にしてプロジェクト定義のリストをソートします。
- p** **chg** サブコマンドの中で使用される場合は、このフラグにより、プロジェクト割り当てでの変更を必要とするプロセス ID のリストを渡します。 **add**、**rm**、および **chattr** の各サブコマンドの中で使用される場合は、このフラグにより、プロジェクト定義が更新される LDAP DN を指定します。**ld** および **unld** サブコマンドの中で使用される場合は、このフラグにより、プロジェクトおよびポリシーが LDAP リポジトリにアップロードされることを指定します。このフラグの引数は、プロジェクトおよびポリシーがアップロードされる DN を示します。
- r** ポリシーを再ロードします。
- s** **projectl chattr agg** サブコマンドで使用され、プロジェクト集約プロパティーを使用可能にします。
- u** **projectl chattr agg** サブコマンドで使用され、プロジェクト集約プロパティーを使用不可にします。



## パラメーター

<i>admpath</i>	管理ポリシー・ファイルを選択する場所からのパス。
<i>appname</i>	必要とされるプロジェクト割り当てリストを持つアプリケーションの、絶対パス。
<i>cmd line</i>	<b>projctl exec</b> コマンドで実行されるコマンドの、絶対パス。
<i>comment</i>	プロジェクト・コメント。
<i>DN</i>	LDAP サーバー上のプロジェクトおよびポリシーのオブジェクトに至る絶対パスを示す識別名。
<i>name</i>	LDAP サーバー上の代替管理ポリシー定義の名前。
<i>pid</i>	プロセス ID。
<i>projname</i>	プロジェクトの名前。
<i>projnumber</i>	プロジェクト用の数値。
<i>projpath</i>	プロジェクト定義ファイルを選択する場所からのパス。
<i>sourceprojpath</i>	マージするプロジェクト定義ファイルを選出する場所からのパス。
<i>targetprojfile</i>	プロジェクト定義がマージされるターゲット・プロジェクト定義ファイル。

## サブコマンド

### add サブコマンド

**add** サブコマンドは、プロジェクトの定義をプロジェクト定義ファイルに追加します。 **-d** フラグが指定されると、プロジェクト定義は、指定されたパスの下のプロジェクト定義ファイルに追加されます。デフォルトでは、**/etc/project/projdef** システム・プロジェクト定義ファイルに追加されます。これ以外のパスの下のプロジェクト定義ファイルは、**.projdef:** という名前にする必要があります。新規プロジェクトがシステム・プロジェクト定義ファイルに追加されるが、プロジェクトがカーネルに既にロードされている場合は、指定された新規プロジェクトはカーネル・プロジェクト・レジストリーに追加されます。それ以外の場合、エントリーはファイル内にのみ作成されます。 **add** サブコマンドは、プロジェクト名、プロジェクト番号、およびプロジェクト・コメント用のオプション引数を、パラメーターとして入手します。デフォルトでは、プロジェクトの集約プロパティーは、このコマンドを使用して作成されたすべてのプロジェクトに対して **no** に設定されます。

**-p** を指定した場合は、新規プロジェクト定義が LDAP サーバー上のデフォルト・プロジェクト *DN* または指定された *DN* に追加されます。 **-p** を指定しない場合は、**.config** がソース情報を提供します。 **-p** オプションの実行には、**root** 権限が必要です。

**projctl add** がプロジェクト定義ファイル内に作成する各エントリーは、以下のフォーマットになります。

```
ProjectName:ProjectNumber:AggregationStatus::Comment
```

ファイル・フォーマットを説明するプロジェクト定義の例は、以下のとおりです。

```
:: Project Definition File
:: Dated: 23-JUN-2003
AIX:3542:yes::To Classify AIX Legacy Applications
Test_Project:0x10000:yes::To Classify Testing work
```

### chattr agg サブコマンド

**chattr agg** サブコマンドは、指定されたプロジェクトの集約プロパティーを使用可能にしたり使用不可にしたりします。 **-s** フラグが使用されると、集約は使用可能になります。 **-u** フラグが使用されると、集約は使用不可になります。 **-d** フラグが指定されると、プロジェクト定義は、指定されたパスの下のプロジェクト定義ファイル内で更新されます。デフォルトでは、システム・プロジェクト定義ファイル (**/etc/project/projdef**) を更新します。更新がシステム・プロジェクト定義ファイルに対するもので、それ

がカーネルに既にロードされている場合は、指定された新規プロジェクトはカーネル・プロジェクト・レジストリーにおいても更新されます。それ以外の場合、変更はプロジェクト定義ファイルに対してのみ行われます。

**-p** を指定した場合は、LDAP サーバー上のデフォルト・プロジェクト *DN* または指定された *DN* に関するプロジェクト定義が変更されます。**-p** を指定しない場合は、**.config** がソース情報を提供します。**-p** オプションの実行には、**root** 権限が必要です。

## chg サブコマンド

**chg** サブコマンドにより、ユーザーは、自分のプロセス用に使用することが許可されているプロジェクトのリストを変更することができます。対象とするプロジェクト名が、このサブコマンドの入力として指定されます。プロセス *ID* が入力として提供された場合は、これらのプロセスは指定されたプロジェクトの下に分類されることになります。入力として提供されたプロセス *ID* がない場合は、プロジェクト変更は、**projctl** コマンドを開始したプロセスに対して行われます。

デフォルトでは、**chg** サブコマンドは、使用可能なルールの有効範囲内でプロジェクト割り当てを変更します。ルールをオーバーライドしてプロジェクトを直接にプロセスに割り当てるには、**-f** 強制オプションを指定する必要があります。

## chk サブコマンド

**chk** サブコマンドは、さまざまなプロジェクト・ポリシーの妥当性を検査します。サブコマンドは、プロジェクトおよびポリシーを安全にカーネルにロードできるように、これらを妥当性検査します。さまざまなプロジェクト・ポリシーをサポートするために、いくつかの **chk** サブコマンドがあります。サブコマンドを以下に示します。

- chkadm** 管理ポリシーを妥当性検査します。管理ポリシー・ファイルの中の各ルールは通常、4 つの属性を持ちます。すなわち、ユーザー *ID*、グループ *ID*、アプリケーション・パス、およびプロジェクト名です。**chkadm** サブコマンドは、これらの属性が妥当か検査し、ポリシーにエラーが見付かれれば報告します。**-d** オプションが使用される場合、**chkadm** サブコマンドは、指定されたパスからの管理ポリシー・ファイルをルールの検査に使用します。また、別名および一時プロジェクト定義ファイル (**.projdef**) も、必要な場合には使用します。ルールで使用されるプロジェクトは、システム・プロジェクト定義ファイルで最初に検索されることになります。プロジェクトがシステム・プロジェクト定義ファイルで見付からない場合は、指定されたパスの下の **.projdef** ファイルが使用されます。
- chkall** 上記の妥当性検査をすべて実行します。つまり、プロジェクト、ユーザー、グループ、および管理ポリシーを妥当性検査します。**-d** オプションが使用される場合、**chkall** サブコマンドは、指定されたパスからの管理、別名、およびプロジェクト定義ファイルを、管理ポリシーの妥当性検査に使用します。
- chkgrp** グループ・ポリシーを妥当性検査します。妥当性検査では、グループのプロジェクト・リストに妥当なプロジェクトが入っているか検査することが必要です。
- chkprojs** システム・プロジェクト定義ファイルを妥当性検査します。プロジェクト定義は、固有性、プロジェクト名と番号の妥当性、および属性の妥当性について、妥当性検査されます。プロジェクト名は POSIX 英数字ストリングであることが必要で、プロジェクト番号は 0x00000001 から 0x00ffffff の範囲内の数字であることが必要です。プロジェクト番号は、10 進数か 16 進数のいずれかが可能です。すべての 16 進数は、プレフィックス 0x を持つことが必要です。集約プロパティは、y または n のいずれかが可能で、集約の状況を示します。**chkprojs** サブコマンドは、プロジェクト定義に対するこれらすべての妥当性検査を実行し、プロジェクト定義にエラーが見付かれれば報告します。
- chkusr** ユーザー・ポリシーを妥当性検査します。妥当性検査では、ユーザーのプロジェクト・リストに妥当なプロジェクトが入っているか検査することが必要です。

注: 管理ポリシー・ルールでワイルドカード文字が使用される場合、**chkadm** サブコマンドおよび **chkall** サブコマンドはワイルドカード文字を展開し、得られた出力を妥当性検査します。

## exec サブコマンド

**exec** サブコマンドは、ユーザーが、コマンドの処理対象となりうるプロジェクトのリストからのプロジェクト名のいずれかを使用して、任意のコマンドを起動することができるようにします。**chg** オプションと同様に、ルールをオーバーライドし任意のプロジェクトを使用してコマンド行を実行するために使用される、**-f** 強制オプションを使用する必要があります。コマンドを割り当てる対象となりうるプロジェクトのリストを取得するには、**projectl qapp** サブコマンドを使用します。

## ld サブコマンド

**ld** サブコマンドは、プロジェクトおよびポリシーのロードと再ロードのために使用されます。特定のポリシーに対してロード操作を行うための、特定のロード・コマンドがあります。このようなさまざまなサブコマンドを、以下に示します。

**ld** システム起動時にロードされる必要のあるポリシーを、ロードします。**/etc/project.config** ファイルを参照して、ロードするポリシーを判断します。既にカーネルに、あるポリシーまたはプロジェクト定義がロードされている場合は、このコマンドは単に戻ります。

## ldadm

管理ポリシーをロードします。 **ldusr** サブコマンドおよび **ldgrp** サブコマンドと同様、**ldadm** も、プロジェクトがまだロードされていない場合は、最初にプロジェクトを検査してロードします。それから、管理ポリシー・ルールを妥当性検査した後でロードします。 **-d** オプションが使用される場合、管理ポリシー・ファイルは指定されたパスから選ばれます。指定されたパスの下の別名および一時プロジェクト定義ファイルは、別名エントリおよびプロジェクト・エントリの存在を検査するために使用されます。ポリシーがロードされた後、このサブコマンドは、管理ポリシー・ファイルを **/etc/project/admin** にコピーすることも行います。LDAP に関連する管理ポリシーのロードは、以下の **-p** および **-g** 引数によって処理されます。

### projectl ldadm -g [name] [name]

LDAP リポジトリを使用して管理ポリシーがカーネルにロードされることを指定します。 **-g** を指定しない場合は、ローカル管理ポリシー (**/etc/project/admin**) がカーネルにダウンロードされます。

### projectl ldadm -g [ [name:]DN | name ] -d admpath

LDAP 管理ポリシーを、カーネルにダウンロードせずに、ローカル・ファイルにダウンロードすることを指定します。ソース管理ポリシーは、指定された DN に置かれ、**ldap.cfg** ファイル内のアカウント DN を使用して検出されます。 **-d** パラメーターを使用して、ポリシー・ファイル (プロジェクト、管理、および別名) が書き込まれる場所を指定します。ターゲット場所が **/etc/project/** の下にある場合は、ファイルは、システムで使用される規則に従って書き込まれます。ファイルは以下のものに書き込まれます。

- **/etc/project/admin**、**/etc/project/alias**、**/etc/project/projdef**
- **/etc/project/ldap/admin**、**/etc/project/ldap/alias**、**/etc/project/ldap/projdef**
- **/etc/project/projdef**、**/etc/project/alter/policyname/admin**、**.../alias**
- **/etc/project/ldap/projdef**、**/etc/project/ldap/alter/policyname/admin**、**.../alias**

それ以外の場合は、この 3 つのファイルは、指定されたディレクトリーに書き込まれます。明示 DN が **-g** オプションを用いて指定された場合は、プロジェクトはダウンロードされません。これは、これらのプロジェクトも別の DN に配置される可能性があるためです。この場合、ユーザーはこれらのプロジェクトを別々にダウンロードする必要があります。

### projectl ldadm -p [ [name:]DN | name ] -d admpath

ディレクトリー **localpath** にある管理ポリシーが LDAP サーバーにアップロードされることを指定します。このコマンドは、**localpath/projdef** 一時プロジェクト定義ファイルで検出されるプロジェクトもアップロードします。

明示 DN が **-p** オプションを用いて指定された場合は、管理ポリシーのみが LDAP サーバーにアップロードされます。これは、プロジェクトが別の DN に配置される可能性があるためです。この場合、ユーザーはそれぞれの **.projdef** ファイルを該当の DN に明示的にアップロードする必要があります。システムは、この DN の ID を認識しません。

**-g** または **-p** 引数を使用する場合は、**-d** 引数を指定する必要があります。 **-r** および **-a** 引数は、**-p** 引数と共に指定することはできません。 **-a** 引数が指定され、**-g** 引数が指定されない場合は、**.config** ファイル内の管理ポリシーがロードされます。 **-r** オプションが使用される場合は、**.active** ファイルを使用して、ロードするポリシーの ID を判別します。 **-r** と **-a** オプションは併用できません。

## ldall

ユーザー、グループ、および管理者の各ポリシーをカーネルにダウンロードします。 **ldusr** および **ldgrp** コマンドと同様に、プロジェクト用のアカウント DN が指定されている場合は、このオプションでは、LDAP プロジェクトのダウンロードを試みます。これは、ユーザーおよびグループ・ポリシーがローカルまたは LDAP ユーザーに個々に関連付けられていないためです。このコマンドは、ローカル管理ポリシーのダウンロードに加えて、構成済み管理 DN を使用してデフォルト管理ポリシーのダウンロードを試みます。

## ldgrp

グループ・プロジェクト・ポリシーをロードします。グループ・プロジェクト・ポリシーがまだロードされていない場合、**ldgrp** サブコマンドは最初にプロジェクトを検査してロードします。それからすべてのグループのプロジェクト・リストの妥当性を検査し、ルールをロードします。

## ldprojs

システム・プロジェクト定義 **/etc/project/projdef** ファイルから、プロジェクト定義をロードします。プロジェクトをロードする前に、ルールの妥当性を検査します。ルールが妥当であれば、そのルールをロードします。

### projctl ldprojs -g

LDAP リポジトリを使用してプロジェクト定義がカーネルにロードされることを指定します。

### projctl ldprojs -p

プロジェクト定義を LDAP サーバーにアップロードすることを指定します。 **-g** および **-p** を指定しない場合は、ローカル定義されたプロジェクト (**/etc/project/projdef**) がカーネルにロードされます。

### projctl ldprojs -g [DM] -d localpdfpath

プロジェクトをカーネルにダウンロードせずに、プロジェクト定義ファイルを LDAP リポジトリからローカル・ファイルにダウンロードすることを指定します。 **-d** 引数を指定しない場合は、プロジェクトは **/etc/project/ldap/projdef** にダウンロードされ、さらにカーネルにダウンロードされます。 **-d** 引数は、ユーザーに、指定された場所にファイルを作成するように指示しますが、そのファイルをカーネルにはダウンロードしないように指示します。この場合、**projdef** ファイルは、**.projdef** ファイルの中ではなく、指定された場所に作成されます。ソース・プロジェクト定義は、指定された DN に配置されます。あるいは、**ldap.cfg** ファイル内の構成済みアカウント DN を使用して、ソース・プロジェクト定義を検出することができます。

### projctl ldprojs -d localpdfpath

ローカル・プロジェクト定義ファイルをカーネルにロードします。

### projctl ldprojs -p [DM] -d localpdfpath

指定されたパスにあるプロジェクト定義が LDAP サーバーにアップロードされることを指定します。プロジェクト定義は、指定されたディレクトリーの **projdef** ファイルの中で使用可能にする必要があります。 **-d** 引数を指定しなければならないのは、**-g** または **-p** がユーザーに、指定された場所にファイルを作成するように指示し、そのファイルをカーネルにはダウンロードしないように指示する場合です。この場合は、**projdef** 引数が使用されます。このようにして、アップロードとダウンロードの操作は、パラメーターの指定に関して対称的なものにすることができます。 **-r** および **-a** 引数は、**-p** 引数と共に指定することはできません。 **-a** 引数が指定され、**-g** 引数が指定されない場合は、**.config** ファイル内のプロジェクト・リポジトリがロードされます。 **-r** オプションが使用される場合は、**.active** ファイルを使用して、ロードするプロジェクト・リポジトリを判別します。 **-r** と **-a** オプションは併用できません。

## ldusr

ユーザー・プロジェクト・ポリシーをロードします。ユーザー・プロジェクト・ポリシーがまだロードされていない場合は、**lduser** サブコマンドは最初にプロジェクトを検査してロードします。それからすべてのユーザーのプロジェクト・リストの妥当性を検査し、ルールをロードします。

## 注:

- **-r** オプションが使用される場合、上記のすべてのサブコマンドは、それぞれのポリシーを再ロードします。 **ld -r** サブコマンドは、カーネルを照会して、ロードされたポリシーの詳細を取得し再ロードします。再ロードするポリシー・ファイルは、**/etc/project/active** ファイルから参照されます。
- **-d** オプションと **-r** オプションの両方を指定して **ldadm** サブコマンドおよび **ldall** サブコマンドが出された場合、**-r** は無視されます。
- すべての **ld** サブコマンドは、ロードされたポリシーの詳細を使用して **/etc/project/active** ファイルを更新します。 **-a** オプションが渡された場合、これらのサブコマンドは、**.active** ファイルを更新するのに加えて、**/etc/project.config** ファイルも更新します。 **/etc/project.config** ファイルは、システム・リポート時に自動的にロードされるポリシーについて、詳細を提供します。



## merge サブコマンド

**merge** サブコマンドは、デフォルトでは、指定されたパスの下のプロジェクト定義ファイルで定義されたプロジェクトを、システム・プロジェクト定義 `/etc/project/projdef` ファイルとマージします。ターゲット・プロジェクト・ファイル名が **-d** オプションを使用して渡される場合は、指定されたパスの下のプロジェクト定義がターゲット・プロジェクト定義ファイルとマージされます。マージ操作は、ターゲット・プロジェクト定義ファイルと指定されたパスの下のプロジェクト定義ファイルとの間で矛盾するエントリーがあると、失敗します。**merge** コマンドは、重複するエントリーがあればスキップして、ターゲット・プロジェクト定義ファイル内の固有なエントリーを維持します。

## qapp サブコマンド

**qapp** サブコマンドは、現行環境内でアプリケーションが切り換える対象となりうるプロジェクトのリストを、表示します。指定されたアプリケーションを開始させるのに使用できるすべてのプロジェクトのリストを、表示します。

## qpolicy サブコマンド

**qpolicy** サブコマンドは、現在ロードされているポリシーを表示します。このコマンドはカーネルを照会して、ロードされたポリシーのタイプに関する情報を取得し、それらを表示します。**-g** を指定した場合は、このコマンドは、LDAP デフォルト管理 DN または指定された DN からポリシーをリストします。

## qproj サブコマンド

**qproj** サブコマンドは、引数として渡されるプロジェクト名の詳細を表示します。引数が渡されない場合は、このサブコマンドは、呼び出しプロセスを割り当てる対象になりうる、システム内のすべてのプロジェクト定義をリストします。表示フォーマットは、**qprojs** サブコマンドと同じになります。

## qprojs サブコマンド

**qprojs** サブコマンドは、カーネル・レジストリー内に現在ロードされているすべてのプロジェクト定義のリストを表示します。**-n** オプションは、プロジェクト名を基にソートされたリストを提供します。表示には、プロジェクト名、プロジェクト番号、およびプロジェクトの集約状況が入っています。

## rm サブコマンド

**rm** サブコマンドは、プロジェクト定義ファイルから、ローカルに定義されたプロジェクトの定義を除去します。**-d** フラグが指定されると、プロジェクト定義は、指定されたパスの下のプロジェクト定義ファイルから除去されます。デフォルトでは、システム・プロジェクト定義ファイル (`/etc/project/projdef`) からプロジェクト定義を除去します。更新がシステム・プロジェクト定義ファイルに対するもので、それがカーネルに既にロードされている場合は、指定されたプロジェクトはカーネル・プロジェクト・レジストリーから除去されます。それ以外の場合、エントリーはファイルからのみ、除去されます。

**-p** を指定した場合は、ソースは、プロジェクト定義が除去される LDAP となります。明示 DN が指定された場合は、プロジェクト定義はその特定の DN から除去されます。DN が渡されない場合は、`ldap.cfg` ファイル内に構成されているデフォルト DN が使用されます。LDAP プロジェクトが現在ロードされている場合は、プロジェクト定義は、カーネル・プロジェクト・レジストリーおよびローカル LDAP プロジェクト・ファイルからも除去されます。それ以外の場合、LDAP リポジトリーのみが更新されます。

注: **-p** と **-d** オプションは併用できません。これらのオプションのいずれも指定されない場合は、ソース情報を提供するために `.config` ファイルが使用されます。このコマンドを実行するには、root 権限が必要です。

## unld サブコマンド

**unld** サブコマンドは、プロジェクト・ポリシーをアンロードするために使用されます。 **ld** サブコマンドと同様に、**unld** サブコマンドは、特定のポリシーをアンロードするために使用されます。このようなさまざまなサブコマンドを、以下に示します。

<b>unldadm</b>	管理ポリシーをアンロードします。
<b>unldall</b>	すべてのロードされたポリシーをアンロードします。
<b>unldgrp</b>	グループ・ポリシーをアンロードします。
<b>unldprojs</b>	プロジェクト定義のみをアンロードします。
<b>unldusr</b>	ユーザー・ポリシーをアンロードします。

### 注:

- これらすべてのサブコマンドは、それぞれのポリシーがアンロードされた後に **.active** ファイルを更新します。
- **-a** オプションが使用される場合、それぞれのポリシーのアンロード状況を使用して **/etc/project.config** ファイルも更新されます。
- **-g** パラメーターは、それぞれの LDAP リポジトリをカーネルからアンロードする必要があることを指定するためのものです。 **-g** が指定されていない場合は、**.active** ファイルの中で指定されているロード済みリポジトリがアンロードされます。
- LDAP サーバーから指定された LDAP リポジトリを除去するには、**-p** オプションを指定する必要があります。
- **unldadm** および **unsubcommand** では、*name* パラメーターは、管理 DN に関する管理ポリシー名を示します。

## 終了状況

0	コマンドは正常に実行されました。
>0	エラーが発生しました。
1	読み取り失敗、書き込み失敗、および malloc 失敗の場合のデフォルト・エラー戻りコード。
2	EINVAL および ENOENT
3	EPERM および EACCES
4	EEXIST

## 例

1. システム・プロジェクト定義ファイルにプロジェクト **newproj** を追加するには、次のように入力します。  

```
projctl add newproj 34 "Test Project"
```
2. パス **/tmp/myproj** の下のプロジェクト定義ファイルから、プロジェクト **test1** を除去するには、次のように入力します。  

```
projctl rm test1 -d /tmp/myproj
```
3. プロジェクト **newproj** の集約状況を使用可能にするには、次のように入力します。  

```
projctl chattr agg newproj -s
```
4. **ps** コマンドをプロジェクト **newproj** の下で実行し、既存のルールをオーバーライドするには、次のように入力します。



```
projectl exec newproj "/usr/bin/ps" -f
```

5. 現在ロードされたポリシーを検索するには、次のように入力します。

```
projectl qpolicy
```

#### 出力:

```
Project definitions are loaded.  
Project definition file name: /etc/project/projdef  
User policies are loaded.
```

6. 管理ポリシーをパス /tmp/myproj からロードするには、次のように入力します。

```
projectl ldadm -d /tmp/myproj
```

7. すべてのプロジェクト・ポリシーを、今、およびシステム・リブート時にアンロードするには、次のように入力します。

```
projectl unldall -a
```

8. 新規プロジェクトを別の DN (ここで DN は ou=projects,ou=aacct,ou=cluster1,cn=aixdata です) 上の LDAP リポジトリに追加するには、次のように入力します。

```
projectl add newproj 34 -p ou=projects,ou=aacct,ou=cluster1,cn=aixdata
```

9. LDAP プロジェクトを、デフォルト DN から **/etc/project/ldap** パスの下のローカル・ファイルにダウンロードするには、次のように入力します。

```
projectl ldprojs -g -d /etc/project/ldap
```

10. デフォルト DN 内のラベル newdef の下に保管されている LDAP 管理ポリシーをカーネルにロードするには、次のように入力します。

```
projectl ldadm -g newdef
```

## 位置

**/usr/bin/projectl**

## ファイル

**/usr/bin/projectl**

**/etc/project/projdef**

**/etc/project/ldap/projdef**

**/etc/project.active**

**/etc/project.config**

**/etc/security/ldap/ldap.cfg**

**projectl** コマンドが入っています。

システム・プロジェクト定義ファイルが入っています。

デフォルトの LDAP プロジェクト定義ファイルが入っています。

現在ロードされたポリシーの状況が入っています。

システム・リブート時にロードされるポリシーの状況が入っています。

拡張アカウントング データを処理するための LDAP クライアント構成の詳細が入っています。

## 関連情報

**acctctl** コマンド、**libaacct.a** ライブラリー・ルーチン。

AIX 5L バージョン 5.3 拡張アカウントング・サブシステム。

---

## prompter コマンド

### 目的

プロンプト・エディターを始動させます。

## 構文

**prompter** [ **-erase** *Character* ] [ **-kill** *Character* ] [ **-prepend** | **-nopprepend** ] [ **-rapid** | **-norapid** ] *File*

## 説明

メッセージ・ハンドラー (MH) パッケージの一部である **prompter** コマンドは、メッセージの入力用にプロンプト・エディターを始動させます。**prompter** コマンドは、ユーザーによって始動されません。

**prompter** コマンドはほかのプログラムによってのみコールされます。

**prompter** コマンドは、*File* パラメーターで指定したファイルをオープンし、To: コンポーネントなどのコンポーネントが空でないかをスキャンし、ブランクのフィールドを埋めるようにユーザーへプロンプトを表示します。必須フィールドに入力せずに Enter キーを押した場合、**prompter** コマンドはそのコンポーネントを削除します。

**prompter** コマンドは、ファイルの最初のブランク行またはダッシュ行の後で、メッセージ本文にテキストを受け入れます。本文に既にテキストが含まれていて、**-nopprepend** フラグが指定されている場合は、**prompter** コマンドは以下のようにテキストに続いてメッセージを表示します。

```
-----Enter additional text
```

**prompter** コマンドは、既存のメッセージの後に入力した新規のテキストを付加します。**-prepend** フラグを指定した場合、**prompter** コマンドは以下のようなメッセージを表示します。

```
-----Enter initial text
```

新規のテキストはどれも、元のメッセージの本文より前に来ます。Ctrl-D キー・シーケンス (ファイル終わり) を押すと、**prompter** コマンドはテキスト入力を終了させ、コール・プログラムへ制御を戻します。

## フラグ

<b>-erase</b> <i>Character</i>	消去文字として使用される文字を設定します。 <i>Character</i> 変数の値には、\NNN フォーマットの 8 進表記の文字を使用できます。 \NNN は数値または文字自体です。例えば、文字 \e は 8 進表記では \145 になります。
<b>-help</b>	コマンド構文、使用可能なスイッチ (トグル)、およびバージョン情報をリストします。
<b>-kill</b> <i>Character</i>	注: MH の場合は、このフラグ名を省略することはできません。 抹消文字または停止文字として使用される文字を設定します。 <i>Character</i> 変数の値には、\NNN フォーマットの 8 進表記の文字を使用できます。 \NNN は数値または文字自体です。例えば、文字 \e は 8 進表記では \145 になります。
<b>-nopprepend</b>	既にメッセージ本文にあるテキストの後に追加テキストを追加します。
<b>-norapid</b>	既にメッセージ本文にあるテキストを表示します。これはデフォルトです。
<b>-prepend</b>	既にメッセージ本文にあるテキストの前に追加テキストを追加します。これはデフォルトです。
<b>-rapid</b>	既にメッセージ本文にあるテキストを表示しません。

## プロファイル・エントリー

Msg-Protect:	新しいメッセージ・ファイルの保護レベルを設定します。
prompter-next:	<b>prompter</b> コマンドを終了した後で使用するエディターを指定します。

## ファイル

`$HOME/mh_profile`

MH ユーザー・プロファイルが入っています。

`/tmp/prompter*`

メッセージの一時コピーが入っています。

## 関連情報

**comp** コマンド、 **dist** コマンド、 **forw** コマンド、 **repl** コマンド、 **whatnow** コマンド。

ネットワークおよびコミュニケーションのマネージ のメール・アプリケーションのセクション。

---

## proto コマンド

### 目的

ファイルシステムのプロトタイプ・ファイルを作成します。

### 構文

**proto** *Directory* [ *Prefix*]

### 説明

**proto** コマンドは、ファイルシステムまたはファイルシステムの一部のためにプロトタイプ・ファイルを作成します。**mkfs** コマンドは、プロトタイプ・ファイルを入力として使用し、事前定義テンプレートに従ってファイルシステムを作成します。プロトタイプ・ファイルは、ファイルシステム上のすべてのファイルの再帰ディレクトリーのリストと、そのオーナー、グループ、および保護で構成されています。また、プロトタイプ・ファイルが初期化され、**mkfs** コマンドに記述されているようにフォーマットされる元となるファイルも入っています。

プロトタイプ・ファイルの作成元となるルート・ディレクトリーを指定するには *Directory* パラメーターを使用します。プロトタイプ・ファイルには、*Directory* パラメーターの下の完全サブツリーが含まれ、プロトタイプ・ファイル自体は *Directory* パラメーターで指定された基本ディレクトリーと同じファイルシステム上に入っています。

*Prefix* パラメーターは、すべての初期設定ファイル名に追加され、初期設定ファイルが強制的にプロトタイプ以外のロケーションから取り出されるようにします。**proto** コマンドからの出力を **mkfs** コマンドとともに使用できるようにするには、**mkfs** コマンドに、始動プログラム、ファイルシステムのサイズ、および *i* ノード・リストのサイズが必要です。**proto** コマンドでは、ハード・リンク情報は保存されません。

照合順序は **LANG** 環境変数または **LC\_COLLATE** 環境変数によって決定されます。

### 例

既存のファイルシステム `/works` にプロトタイプ・ファイルを作成するには、以下のように入力します。

```
proto /works
```

`/works` ファイルシステムに 2 つのディレクトリー `dir1` および `dir2` があり、ディレクトリー `dir1` にファイル `file1` がある場合、**proto** コマンドは以下のように表示します。

```
#Prototype file for /works
d--- 755 0 0
  dir1 d--- 755 0 0
    file1 ---- 644 0 0 /works/dir1/file1
      $
  dir2 d--- 755 0 0
    $
  $
$
```

## ファイル

`/usr/sbin/proto` `proto` コマンドが入っています。

## 関連情報

`mkfs` コマンド、`mkproto` コマンド。

オペレーティング・システムおよびデバイスのマネージのファイルシステムのセクションは、ファイルシステムのタイプ、管理、構造、および保守について説明しています。

オペレーティング・システムおよびデバイスのマネージのファイルのセクションでは、ファイルの取り扱いについての情報が提供されています。

*AIX 5L* バージョン 5.3 ナショナル・ランゲージ・サポート ガイドおよびリファレンスのナショナル・ランゲージ・サポートの概要のセクション。

*AIX 5L* バージョン 5.3 ナショナル・ランゲージ・サポート ガイドおよびリファレンスのロケールのセクション。

---

## proxymngr コマンド

### 目的

プロキシー・マネージャー・サービス。

### 構文

```
proxymngr [ -config filename] [ -timeout seconds] [ -retries #] [ -verbose]
```

### 説明

`proxymngr` (プロキシー・マネージャー) は、`xfindproxy` (およびその他の同様のクライアント) からの要求を解決し、適切であれば新しいプロキシーを始動し、さらに使用可能な全プロキシー・サービスをトラッキングする役目を果たします。プロキシー・マネージャーは、できるだけ既存のプロキシーを再利用しようとしています。

プロキシー・マネージャーが取り扱うプロキシーには、管理プロキシーと非管理プロキシーの 2 つのタイプがあります。

管理プロキシーは、プロキシー・マネージャーの要求時に始動されるプロキシーです。

非管理プロキシは、システム・ブート時に始動されるか、システム管理者によって手動で始動されます。プロキシ・マネージャーにはその存在は知らされますが、プロキシ・マネージャーが非管理プロキシを始動しようとすることはありません。

## フラグ

<b>-config</b>	デフォルト <b>proxymngr config</b> ファイルをオーバーライドします。 <b>proxymngr config</b> ファイルの詳細は、以下を参照してください。
<b>-timeout</b>	非管理プロキシを見つけるためのプロキシ・マネージャーの試みから次の試みまでの秒数を設定します。デフォルトは 10 です。
<b>-retries</b>	非管理プロキシを見つけるためにプロキシ・マネージャーが再試行する最大回数を設定します。デフォルトは 3 です。
<b>-verbose</b>	要求が受信され、プロキシが始動されると、さまざまなデバッグ・レコードとトレース・レコードが表示されます。

## プロキシ・マネージャー Config ファイル

プロキシ・マネージャーは、使用可能なプロキシ・サービスを記述するローカル構成ファイルを維持します。この構成ファイルは、**proxymngr** のインストール時に **/usr/X11R6.3/lib/X11/proxymngr/pmconfig** にインストールされます。この構成ファイルの位置は、**-config** コマンド行フラグを使って上書きできません。

コメントの感嘆符から開始される行とは別に、構成ファイルの各行は、非管理プロキシ・サービスまたは管理プロキシ・サービスを記述します。

非管理プロキシの場合のフォーマットは次のとおりです。

```
<service-name> unmanaged <proxy-address>
```

*service-name* は、非管理プロキシ・サービスの名前で、スペースを入れてはなりません。例えば、XFWP のようにします。 *service-name* は大/小文字を区別しません。

*proxy-address* は非管理プロキシのネットワーク・アドレスです。このアドレスのフォーマットは、*service-name* に特定されたものです。例えば、XFWP サービスの場合、*proxy-address* は `firewall.x.org:100` などというようになります。

同じ非管理 *service-name* を持つ `config` ファイル内に複数のエントリーがある場合、プロキシ・マネージャーは、その `config` ファイルに表示される順序でそれらのプロキシを使用しようとします。

管理プロキシの場合、フォーマットは次のとおりです。

```
<service-name> managed <command-to-start-proxy>
```

*service-name* は、管理プロキシ・サービスの名前で、スペースを入れてはなりません。例えば、LBX のようにします。 *service-name* は大/小文字を区別しません。

*command-to-start-proxy* は、プロキシ・マネージャーによって、プロキシの新しいインスタンスを始動するために実行されるコマンドです。 *command-to-start-proxy* にスペースが含まれている場合、そのコマンド全体を単一引用符で囲む必要があります。必要なら、*command-to-start-proxy* を使用して、リモート・マシン上のプロキシが始動できます。これを行うために使用するリモート実行メソッドの詳細はここでは紹介しません。

例: 構成ファイルの例

```
! proxy manager config file
!
! Each line has the format:
!   <serviceName> managed <startCommand>
!     or
!   <serviceName> unmanaged <proxyAddress>
!
! lbx managed /usr/X11R6.3/bin/lbxproxy
!
! substitute site-specific info
xfwp unmanaged firewall:4444
```

## プロキシー・マネージャーの詳細

プロキシー・マネージャーが **xfindproxy** (または他の同様のクライアント) から要求を受け取ったときの一連のアクションは、該当する *service-name* によって異なります。

管理プロキシー・サービスの場合、プロキシー・マネージャーは、そのサービスに関して既に実行しているプロキシーのいずれかが、新しい要求を処理できるかどうかを確認します。どのプロキシーもその処理ができない場合、プロキシー・マネージャーは、*config* ファイル内にある *command-to-start-proxy* を使用して、プロキシーの新しいインスタンスの始動を試みます。その試みに失敗すると、エラーがコール元に戻されます。

非管理プロキシー・サービスの場合、プロキシー・マネージャーは、*config* ファイルを調べて、このサービス用の非管理プロキシーをすべて見つけます。同じ非管理 *service-name* を持つ *config* ファイル内に複数のエントリーがある場合、プロキシー・マネージャーは、その *config* ファイルに表示される順序でそれらのプロキシーを使用しようとします。非管理プロキシーがどれもその要求に応えることができない場合、プロキシー・マネージャーは、(**-timeout** またはデフォルトの 10 で指定された) 構成可能な時間の間だけタイムアウトになり、その要求に応じる意志のある非管理プロキシーの検出を再度試みます。再試行の回数は、**-retries** 引数によって指定でき、場合によってはデフォルトの 3 を使用します。再試行に失敗した場合、プロキシー・マネージャーに残された選択は、コール元にエラーを戻すことだけです (プロキシー・マネージャーは非管理プロキシー・サービスは始動できないためです)。

---

## prs コマンド (SCCS)

### 目的

ソース・コード制御システム (SCCS) のファイルを表示します。

### 構文

```
prs [ -a ] [ -d String ] [ -r [ SID ] ] [ -c Cutoff ] [ -e | -l ] File ...
```

### 説明

**prs** コマンドは、まず指定ファイルを読み取り、次にソース・コード制御システム (SCCS) ファイルの一部またはすべてを標準出力へ書き出します。*File* パラメーターにディレクトリーを指定すると、**prs** コマンドはすべての SCCS ファイル (**s.** 接頭部の付いたファイル) で要求されたアクションを実行します。*File* パラメーターに **-** (マイナス) を設定すると、**prs** コマンドは標準入力を読み取り、各行を SCCS ファイル名として解釈します。**prs** コマンドは、ファイル終わり文字に達するまで、入力の読み取りを続けます。

## データ・キーワード

データ・キーワードは SCCS ファイルの中で、検索されて標準出力に書き出される部分を指定します。SCCS ファイルのすべての部分に、関連するデータ・キーワードがあります。データ・キーワードは指定された指定ファイル内で、何度でも使用できます。

**prs** コマンドが表示する情報は、ユーザー提供のテキスト、および認識されたデータ・キーワードが該当する値 (SCCS ファイルから抽出された値) で置き換えられて、指定されたファイルに表示される順番で示されます。データ・キーワード値のフォーマットは、キーワード置き換えが直接行われる単純フォーマットか、または置き換えの後に復帰が続く複数行フォーマットです。テキストは、認識されたデータ・キーワード以外の文字で構成されます。**\t** (バックスラッシュ、文字 t) でタブ文字を、**\n** (バックスラッシュ、文字 n) で復帰または改行文字を指定します。**\t** および **\n** に余分な **\** (バックスラッシュ) を付けると、シェルが **\** を解釈しなくなり、文字 **t** または **n** のみをテキストとして **prs** コマンドに渡すので注意してください。

以下の表は、SCCS ファイルのデルタ・テーブルの情報と関連したキーワードをリストしたものです。特に断りがない限り、すべてのキーワードは単純フォーマットです。

デルタ・テーブルのキーワード

キーワード	表されるデータ	値
<b>:R:</b>	リリース番号	数値
<b>:L:</b>	レベル番号	数値
<b>:B:</b>	分岐番号	数値
<b>:S:</b>	シーケンス番号	数値
<b>:I:</b>	SCCS の ID 文字列 (SID)	<b>:R::L::B::S:</b>
<b>:Dy:</b>	デルタ作成年	YY
<b>:Dm:</b>	デルタ作成月	MM
<b>:Dd:</b>	デルタ作成日	DD
<b>:D:</b>	デルタ作成日付	YY/MM/DD
<b>:Th:</b>	デルタ作成時	HH
<b>:Tm:</b>	デルタ作成分	MM
<b>:Ts:</b>	デルタ作成秒	SS
<b>:T:</b>	デルタ作成時刻	HH/MM/SS
<b>:DT:</b>	デルタ・タイプ	D または R

<b>:P:</b>	デルタ作成者	ログイン名
<b>:DS:</b>	デルタ・シーケンス番号	数値
<b>:DP:</b>	直前のデルタ・シーケンス番号	数値
<b>:Dt:</b>	デルタ情報	<b>:DT::I::D::T::P::DS::DP:</b>
<b>:Dn:</b>	含まれるデルタのシーケンス番号	<b>:DS: . . .</b>
<b>:Dx:</b>	除外されるデルタのシーケンス番号	<b>:DS: . . .</b>
<b>:Dg:</b>	無視されるデルタのシーケンス番号	<b>:DS: . . .</b>
<b>:DI:</b>	含まれ、除外され、無視されるデルタのシーケンス番号	<b>:Dn::/Dx::/Dg:</b>
<b>:Li:</b>	デルタが挿入する行数	数値
<b>:Ld:</b>	デルタが削除する行数	数値



<b>:Lu:</b>	デルタが変更しない行数	数値
<b>:DL:</b>	デルタ行の統計情報	:Li:/Ld:/Lu:
<b>:MR:</b> (複数行フォーマット)	デルタの MR 番号	テキスト
<b>:C:</b> (複数行フォーマット)	デルタ用のコメント	テキスト

以下の表は、SCCS ファイルのヘッダー・フラグと関連したキーワードをリストしたものです。特に断りがない限り、すべてのキーワードは単純フォーマットです。

ヘッダー・フラグのキーワード

キーワード	表されるデータ	値
<b>:Y:</b>	モジュール・タイプ	テキスト
<b>:MF:</b>	MR 妥当性検査フラグ・セット	yes または no
<b>:MP:</b>	MR 妥当性検査プログラム名	テキスト
<b>:KF:</b>	キーワード/エラー警告フラグのセット	yes または no
<b>:BF:</b>	分岐フラグのセット	yes または no
<b>:J:</b>	結合編集フラグのセット	yes または no
<b>:LK:</b>	ロック・リリース	:R: . . .
<b>:Q:</b>	ユーザー定義のキーワード	テキスト
<b>:M:</b>	モジュール名	テキスト
<b>:FB:</b>	フロア境界	:R:
<b>:CB:</b>	シーリング境界	:R:
<b>:Ds:</b>	デフォルト SID	:I:
<b>:ND:</b>	null デルタ・フラグ・セット	yes または no
<b>:FL:</b> (複数行フォーマット)	ヘッダー・フラグ・リスト	テキスト

以下の表は、SCCS ファイルのほかの部分と関連したキーワードをリストしたものです。特に断りがない限り、すべてのキーワードは単純フォーマットです。

その他のキーワード

キーワード	表されるデータ	値
<b>:UN:</b> (複数行フォーマット)	ユーザー名	テキスト
<b>:FD:</b> (複数行フォーマット)	記述テキスト	テキスト
<b>:BD:</b> (複数行フォーマット)	テキストの本体	テキスト
<b>:GB:</b> (複数行フォーマット)	g ファイル内のテキスト	テキスト
<b>:W:</b>	what 文字列	:Z::M: \tab :I:
<b>:A:</b>	what 文字列	:Z::Y::M::I::Z:
<b>:Z:</b>	what 文字列の区切り文字	@(#)
<b>:F:</b>	SCCS ファイル名	テキスト
<b>:PN:</b>	SCCS ファイルのパス名	テキスト

## フラグ

指定された各ファイルには、各フラグまたはフラグ・グループが個別に適用されます。

- a** 指定デルタの情報を、除去されているか否かに関係なく書き込みます (**rmdel** コマンドを参照してください)。 **-a** フラグを指定しない場合には、**prs** コマンドは除去されていない指定デルタについてのみ情報を提供します。
- c Cutoff** **-e** および **-l** フラグに対して、カットオフの日付と時刻を指定します。以下のフォーマットで、*Cutoff* 値を指定します。  
YY[MM[DD[HH[MM[SS]]]]]  
  
省略されたエンタリーにはすべてデフォルト値として最大値が使用されるので、**-c8402** と指定するのは **-c840229235959** と指定するのと同じこととなります。フィールドは、非数値文字で区切ることができます。例えば、**-c84/2/20,9:22:25** または **-c"84/2/20 9:22:25"** または **"-c84/2/20 9:22:25"** と指定できます。**-c** フラグは **-r** フラグとともに指定することはできません。
- d String** 表示されるデータ・エンタリーを指定します。文字列は、オプションのテキストおよび SCCS ファイル・データ・キーワードで構成されます。文字列には MBCS (マルチバイト文字セット) 文字を含めることができます。文字列にスペースが入っている場合、その文字列は引用符で囲まなければなりません。
- e** **-r** フラグで指定されたデルタを含めて、それ以前に作成されたすべてのデルタに関する情報を要求します。
- l** **-r** フラグで指定されたデルタを含めて、それ以降に作成されたすべてのデルタに関する情報を要求します。
- r[SID]** **prs** コマンドが情報を検索するデルタの SCCS ID 文字列 (SID) を指定します。**-r** フラグとオプションの SID パラメーターの間には、スペースを入れないでください。SID が指定されない場合は、このコマンドは最高番号のデルタの SID に関する情報を検索します。**-r** フラグは **-c** フラグとともに指定することはできません。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0** 正常終了。
- >0** エラーが発生しました。

## 例

1. SCCS ファイル名 **s.test.c** について作成されたすべてのデルタ (**rmdel** コマンドを使用して除去したすべてのデルタを含む) に関する情報を表示するには、以下のように入力します。

```
prs -a s.test.c
```

2. **s.test.c** の SID 1.2 に関して、ユーザー・ログイン名、デルタによって挿入された行数、およびデルタによって削除された行数を表示するには、以下のように入力します。

```
prs -r1.2 -d":P:\n:Li:\n:Ld:" s.test.c
```

## ファイル

**/usr/bin/prs** **prs** コマンドが入っています。

## 関連情報

**admin** コマンド、 **delta** コマンド、 **get** コマンド、 **sccshelp** コマンド。

**sccsfile** ファイル・フォーマット。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」のソース・コード制御システム (SCCS) の概要のセクション。

---

## prtacct コマンド

### 目的

**tacct** フォーマットのファイルをフォーマットし、表示します。

### 構文

```
/usr/sbin/acct/prtacct [ -X ] [ -W ] [ -f Fields ] [ -v ] File [ "Heading" ]
```

### 説明

**prtacct** コマンドは、合計アカウントリング・ファイルをフォーマットして表示します。これらのファイルは **tacct** フォーマットです。このコマンドを入力すると、接続時間、処理時間、ディスク使用状況、およびプリンター使用状況に関する日次レポートなどの任意の **tacct** ファイルを表示できます。**Heading** パラメーターでレポートのタイトルを指定するには、見出しテキストを " " (引用符) で囲みます。

### フラグ

<b>-f Fields</b>	<b>acctmerg</b> コマンドのフィールド選択メカニズムを使用して、表示するフィールドを選択します。
<b>-v</b>	浮動小数点に対してより正確な表記が使用されるように、詳細出力を作成します。
<b>-W</b>	最初の 8 文字に切り捨てることをせずに、各ユーザー名の使用可能なすべての文字を処理します。またこのフラグを使用すると、 <b>prtacct</b> コマンドが、 <b>tacctx</b> 構造で読み取ることを想定します。これにより、このコマンドは同じ列順序で出力しますが、長いユーザー名での列の位置のずれを許容するようになります。 <b>-W</b> フラグと <b>-X</b> フラグを一緒に使用すると、 <b>-X</b> が優先されます。
<b>-X</b>	最初の 8 文字に切り捨てることをせずに、各ユーザー名の使用可能なすべての文字を処理します。またこのフラグを使用すると、 <b>prtacct</b> コマンドが、 <b>tacctx</b> 構造で読み取ることを想定し、ユーザー名を最後の列に出力します。 <b>-W</b> フラグと <b>-X</b> フラグを一緒に使用すると、 <b>-X</b> が優先されます。

### セキュリティ

アクセス制御: このコマンドは、**adm** グループのメンバーのみに実行 (x) アクセス権限を与えます。

### 例

接続時間のアカウントリング用に合計アカウントリング・ファイルから選択したレコードをフォーマットして表示するには、最初に **prtacct** コマンドの実行対象となるファイルを作成しなければなりません。この例では、**acctcon1** コマンドおよび **acctcon2** コマンドを使用して、**tacct** ファイルを作成します。以下のように入力します。

```
tail /var/adm/wtmp > wtmp.sav
```

```
acctcon1 -t < wtmp.sav | sort +1n +2 | acctcon2 > tacct
```

接続時間アカウンティング・データを処理するためにこのファイルをあらかじめ作成しておく、そのファイルをあらためて作成し直す必要はありません。

次に、**-f** フラグを指定した **prtacct** コマンドを使用して、表示したい合計アカウンティング・ファイルのデータのフィールドを表示します。見出し用のテキストは、引用符で囲むことができます。ログイン名、基本接続時間、および非基本接続時間のレコードを表示し、**Connect-time Accounting** という見出しを付けるには、以下のように入力します。

```
prtacct -f 2,11,12 tacct "Connect-time Accounting"
```

このコマンドを入力すると、処理時間、ディスク使用状況、およびプリンター使用状況に関する日次レポートなどのほかの合計アカウンティング・ファイルをフォーマットして表示することもできます。

## ファイル

<b>/usr/sbin/acct</b>	アカウンティング・コマンドへのパス。
<b>/var/adm/pacct</b>	プロセス・アカウンティング用の現行ファイル。
<b>/var/adm/pacct*</b>	<b>pacct</b> ファイルが大きくなり過ぎた場合に使用されます。

## 関連情報

**acctcon1** または **acctcon2** コマンド、**acctdisk** または **acctdusg** コマンド、**acctmerg** コマンド、**acctprc1**、**acctprc2**、または **accton** コマンド。

**acct** ファイル・フォーマットの **tacct** 構造体。

アカウンティング・システム、日次および月次のレポートの準備、およびアカウンティング・ファイルについての詳細は、オペレーティング・システムおよびデバイスのマネージのシステム・アカウンティングのセクションを参照してください。

オペレーティング・システムおよびデバイスのマネージのアカウントング・サブシステムの設定のセクションで、アカウントング・システムの確立のための必要なステップが説明されています。

---

## prtconf コマンド

### 目的

システム構成情報を表示します。

### 構文

```
prtconf [ -c ] [ -k ] [ -L ] [ -m ] [ -s ] [ -v ]
```

### 説明

フラグを指定せずに **prtconf** コマンドを実行すると、システム・モデル、マシン・シリアル番号、プロセッサ・タイプ、プロセッサの数、プロセッサ・クロック速度、**cpu** タイプ、合計メモリー・サイズ、ネットワーク情報、ファイルシステム情報、ページング・スペース情報、およびデバイス情報が表示されます。

## フラグ

-c	cpu タイプ (例: 32 ビットまたは 64 ビット) を表示します。
-k	使用中のカーネル (例: 32 ビットまたは 64 ビット) を表示します。
-L	LPAR 区画番号および、これが LPAR 区画の場合は区画名を表示し、そうでない場合は "-1 NULL" を戻します。
-m	システム・メモリーを表示します。
-s	プロセッサのクロック速度をメガヘルツで表示します。
-v	デバイスのカスタマイズされた VPD オブジェクト・クラスに見つかった VPD を表示します。

## 終了状況

- 0 コマンドは正常に実行されました。
- >0 エラーが発生しました。

## 例

1. システム構成情報を表示するには、次のように入力します。

```
prtconf
```

システムは下記のようなメッセージを表示します。

```
System Model: IBM,7025-F50
Machine Serial Number: 1025778
Processor Type: PowerPC_604
Number Of Processors: 2
Processor Clock Speed: 332 MHz
CPU Type: 32-bit
Kernel Type: 32-bit
LPAR Info: -1 NULL
Memory Size: 512 MB
Good Memory Size: 512 MB
Firmware Version: IBM,L02113
Console Login: enable
Auto Restart: false
Full Core: false
```

```
Network Information
Host Name: vd01.austin.ibm.com
IP Address: 9.3.207.112
Sub Netmask: 255.255.255.128
Gateway: 9.3.207.1
Name Server: 9.3.199.2
Domain Name: austin.ibm.com
Paging Space Information
```

```
Total Paging Space: 512MB
Percent Used: 1%
```

```
Volume Groups Information
```

```
=====
rootvg:
PV_NAME          PV STATE          TOTAL PPs   FREE PPs   FREE DISTRIBUTION
hdisk0           active            537         394         107..43..29..107..108
=====
```

## INSTALLED RESOURCE LIST

The following resources are installed on the machine.  
+/- = Added or deleted from Resource List.  
\* = Diagnostic support not available.

Model Architecture: chrp  
Model Implementation: Multiple Processor, PCI bus

+ sys0 00-00 System Object  
+ sysplanar0 00-00 System Planar  
+ mem0 00-00 Memory  
etc.

2. プロセッサのクロック速度を表示するには、次のように入力します。

```
prtconf -s
```

システムは下記のようなメッセージを表示します。

```
Processor Clock Speed: 332 MHz
```

3. カスタマイズされたデータベース内のすべての物理デバイスの VPD を表示するには、次のように入力します。

```
prtconf -v
```

システムは下記のようなメッセージを表示します。

## INSTALLED RESOURCE LIST WITH VPD

The following resources are installed on your machine.

Model Architecture: chrp  
Model Implementation: Uni-Processor, PCI bus

sys0	P1-C1	System Object
sysplanar0		System Planar
mem0		Memory
L2cache0		L2 Cache
proc0	P1-C1	Processor

Device Specific.(YL).....P1-C1

pci0	P1	PCI Bus
------	----	---------

Device Specific.(YL).....P1

isa0	P1	ISA Bus
------	----	---------

Device Specific.(YL).....P1

fda0	P1/D1	Standard I/O Diskette Adapter
------	-------	-------------------------------

Device Specific.(YL).....P1/D1

fd0	P1-D1	Diskette Drive
siokma0	P1/K1	Keyboard/Mouse Adapter

Device Specific.(YL).....P1/K1

sioka0	P1-K1	Keyboard Adapter
kbd0	P1-K1-Lkbd	PS/2 keyboard
sioma0	P1-O1	Mouse Adapter
mouse0	P1-O1-Lmouse3	button mouse
siota0	P1/Q1	Tablet Adapter

Device Specific.(YL).....P1/Q1

```

paud0          P1/Q2      Ultimedia Integrated Audio
    Device Specific.(YL).....P1/Q2

ppa0           P1/R1      CHR P IEEE1284 (ECP) Parallel Port Adapter
    Device Specific.(YL).....P1/R1

sa0            P1/S1      Standard I/O Serial Port
    Device Specific.(YL).....P1/S1

tty0          P1/S1-L0    Asynchronous Terminal
sa1           P1/S2      Standard I/O Serial Port
    Device Specific.(YL).....P1/S2

ent0          P1/E1      IBM 10/100 Mbps Ethernet PCI Adapter (23100020)
    Network Address.....0004AC2A0419
    Displayable Message.....PCI Ethernet Adapter (23100020)
    Device Specific.(YL).....P1/E1

scsi0         P1/Z1      Wide/Fast-20 SCSI I/O Controller
    Device Specific.(YL).....P1/Z1

cd0           P1/Z1-A3    SCSI Multimedia CD-ROM Drive (650 MB)
    Manufacturer.....IBM
    Machine Type and Model.....CDRM00203
    ROS Level and ID.....1_00
    Device Specific.(Z0).....058002028F000018
    Part Number.....97H7608
    EC Level.....F15213
    FRU Number.....97H7610

hdisk0        P1/Z1-A5    16 Bit SCSI Disk Drive (4500 MB)
    Manufacturer.....IBM
    Machine Type and Model.....DDRS-34560W
    FRU Number.....83H7105
    ROS Level and ID.....53393847
    Serial Number.....RDHW5008
    EC Level.....F21433
    Part Number.....03L5256
    Device Specific.(Z0).....000002029F00003A
    Device Specific.(Z1).....00K0159S98G
    Device Specific.(Z2).....0933
    Device Specific.(Z3).....0299
    Device Specific.(Z4).....0001
    Device Specific.(Z5).....22
    Device Specific.(Z6).....F21390

b10           P1.1-I2/G1  GXT255P Graphics Adapter
    GXT255P 2D Graphics Adapter:
    EC Level.....E76756
    FRU Number.....93H6267
    Manufacture ID.....IBM053
    Part Number.....93H6266
    Serial Number.....88074164
    Version.....RS6K
    Displayable Message.....GXT255P
    ROM Level.(alterable).....02
    Product Specific.(DD).....00

```



```
Product Specific.(DG).....00
Device Specific.(YL).....P1.1-I2/G1
```

```
pci1          P1.1          PCI Bus
```

```
Device Specific.(YL).....P1.1
```

4. 使用中のカーネル・タイプを表示するには、次のように入力します。

```
prtconf -k
```

システムは、カーネル・タイプの情報を次のように表示します。

```
Kernel Type: 32-bit
```

5. メモリーを表示するには、次のように入力します。

```
prtconf -m
```

システムはメモリーを次のように表示します。

```
Memory Size: 512 MB
```

## ファイル

`/usr/sbin/prtconf`

`prtconf` コマンドが入っています。

## 関連情報

`lsattr` コマンド、`lsconn` コマンド、`lsdev` コマンド、`lsparent` コマンド、`lscfg` コマンド。

---

## ps コマンド

### 目的

プロセスの状況を表示します。この資料では、標準 AIX `ps` コマンドおよび `ps` コマンドの System V バージョンを説明します。

### 構文

#### X/Open 標準

```
ps [ -A ] [ -M ] [ -N ] [ -Z ] [ -a ] [ -d ] [ -e ] [ -f ] [ -k ] [ -l ] [ -F format ] [ -o Format ] [ -c Clist ] [ -G Glist ] [ -g Glist ] [ -m ] [ -n NameList ] [ -p Plist ] [ -P ] [ -t Tlist ] [ -U Ulist ] [ -u Ulist ] [ -T pid ] [ -L pidlist ] [ -X ]
```

#### Berkeley 標準

```
ps [ a ] [ c ] [ e ] [ ew ] [ eww ] [ ewww ] [ g ] [ n ] [ U ] [ w ] [ x ] [ l | s | u | v ] [ t tty ] [ X ] [ ProcessNumber ]
```

### 説明

`ps` コマンドは、アクティブ・プロセスの状況、および `-m` フラグが指定されていれば、それに関連するカーネル・スレッドを、標準出力に書き出します。`-m` フラグを指定するとプロセスに関連するスレッドに関する行が追加表示されますが、追加のスレッド関連の欄を表示するには、`-o` フラグを **THREAD** フィールド指定子とともに使用しなければなりません。

フラグが指定されていないければ、**ps** コマンドは現行端末に関する情報を表示します。フラグ **-f**、**-o**、**l**、**-l**、**s**、**u**、および **v** は、プロセスに関して表示する情報量のみを決定し、どのプロセスをリストするかを決定するわけではありません。フラグ **l**、**s**、**u**、および **v** は相互に排他的です。

**-o** フラグでは、**ps** コマンドはメモリーまたはページング域を調べて、プロセスが作成されたときのコマンド名およびパラメーターを判別します。**ps** コマンドがこの情報を見つけられなければ、カーネル内に格納されたコマンド名が大括弧で囲まれて表示されます。

**COLUMNS** 環境変数は、システムで選択された横方向の画面サイズをオーバーライドします。

パラメーター・リストを受け入れるコマンド行フラグ (**-o**、**-G**、**-g**、**-p**、**-t**、**-U**、および **-u** フラグ) は、128 エントリーまでに制限されています。例えば、**-u Ulist** フラグに指定できるユーザー数は 128 以内です。

**ps** コマンドに指定したフラグに応じて、標準出力に表示される情報の上に列見出しが表示されます。各見出しの定義を以下にリストします (これらの見出しを表示させるフラグを括弧内に示します)。

**ADDR** (**-l** および **l** フラグ) 通常の場合は、プロセス・スタックのセグメント番号が入っています。カーネル・プロセスの場合は、プリプロセス・データ領域のアドレスです。

**BND** (**-o THREAD** フラグ) カーネル・スレッドがバインドされる論理プロセッサ番号 (バインドされる場合)。プロセスの場合は、そのすべてのスレッドが同じプロセッサにバインドされたときに、このフィールドが表示されます。

**C** (**-f**、**l**、および **-l** フラグ) プロセスまたはスレッドの CPU 使用状況。この値は、システム・クロックが時を刻むごとに増加し、プロセスまたはスレッドが実行中であることがわかります。スケジューラーは、毎秒一度ずつ 2 で除算してこの値を減分させます。sched\_other ポリシーでは、CPU 使用率はプロセスのスケジューリング優先順位を決定する際に使用されます。大きな値は CPU 処理中心のプロセスを示し、プロセス優先順位が低くなりますが、小さな値は入出力処理中心のプロセスを示し、優先順位がそれより高くなります。

**CMD** (**-f**、**-l**、および **l** フラグ) コマンド名が入っています。**-f** フラグのもとでは、**ps** は現在のコマンド名と引数の判別を試みます。これらはいずれもプロセスによって非同期的に変更される場合があります。この後そのコマンド名と引数が表示されます。この表示に失敗すると、**-f** オプションなしにコマンド名が表示されるように、大括弧の中にそのコマンド名が書き込まれます。

## COMMAND

(**s**、**u**、および **v**) コマンド名が入っています。完全なコマンド名およびそのパラメーターは、**-f** フラグで表示されます。

### F フィールドの表

フラグ	16 進数	定義
<b>SLOAD</b>	0x00000001	プロセスがコア・メモリーで動作中であることを示す。
<b>SNOSWAP</b>	0x00000002	プロセスをスワップアウトできないことを示す。
<b>STRC</b>	0x00000008	プロセスがトレースされていることを示す。
<b>SWTED</b>	0x00000010	トレース中に、プロセスが停止したことを示す。
<b>SFWTED</b>	0x00000020	トレース中に、 <b>fork</b> サブルーチンのコール後でプロセスが停止したことを示す。
<b>SEWTED</b>	0x00000040	トレース中に、 <b>exec</b> サブルーチンのコール後でプロセスが停止したことを示す。

F フィールドの表

フラグ	16 進数	定義
<b>SLWTED</b>	0x00000080	トレース中に、 <b>load</b> または <b>unload</b> サブルーチンのコール後でプロセスが停止したことを示す。
<b>SFIXPRI</b>	0x00000100	<b>pcpu</b> フィールド・ディスクリプターを無視し、プロセスが固定優先順位を持つことを示す。
<b>SKPROC</b>	0x00000200	カーネル・プロセスを示す。
<b>SOMASK</b>	0x00000400	シグナルを受信した後で、古いマスクの復元を示す。
<b>SWAKEONSIG</b>	0x00000800	シグナルが <b>sleep</b> サブルーチンを異常終了することを示す。その内容は、 <b>PCATCH</b> フラグの内容に等しいものであってはならない。 <b>PCATCH</b> および <b>SWAKEONSIG</b> の両方の内容が、 <b>PMASK</b> の内容よりも大きくなければならない。
<b>SUSER</b>	0x00001000	プロセスがユーザー・モードにあることを示す。
<b>SLKDONE</b>	0x00002000	プロセスがロックを行ったことを示す。
<b>STRACING</b>	0x00004000	プロセスがデバッグ・プロセスであることを示す。
<b>SMPTRACE</b>	0x00008000	マルチプロセス・デバッグを示す。
<b>SEXIT</b>	0x00010000	プロセスが終了していることを示す。
<b>SSEL</b>	0x00020000	プロセッサが、ウェイクアップ/待機の危険のどちらかを選択していることを示す。
<b>SORPHANPGRP</b>	0x00040000	孤立プロセス・グループを示す。
<b>SNOCNTLPROC</b>	0x00080000	セッション・リーダーを制御端末から解放したことを示す。
<b>SPPNOCLDSTOP</b>	0x00100000	子プロセスが停止するときに親プロセスへ <b>SIGHLD</b> シグナルが送信されないことを示す。
<b>SEXECED</b>	0x00200000	プロセスが稼働したことを示す。
<b>SJOBSESS</b>	0x00400000	現行セッションでジョブ制御が使われたことを示す。
<b>SJOBUFF</b>	0x00800000	プロセスがジョブ制御を受けていないことを示す。
<b>PSIGDELIVERY</b>	0x01000000	プロセスがプログラム・チェック・ハンドラーによって使用されることを示す。
<b>SRMSHM</b>	0x02000000	<b>exit</b> サブルーチンのコール中、プロセスが共用メモリーを除去したことを示す。
<b>SSLOTFREE</b>	0x04000000	プロセスのロットが空いていることを示す。
<b>SNOMSG</b>	0x08000000	これ以上 <b>uprintf</b> サブルーチン・メッセージがないことを示す。

**DPGSZ**

(Z フラグ) プロセスのデータ・ページ・サイズ。

**F** (-I および I フラグ) プロセスおよびスレッドに関連した、より重要な F フィールド・フラグ (16 進数および追加情報) の一部を次の表にリストします。

F フィールドの表

フラグ	16 進数	定義
<b>SLOAD</b>	0x00000001	プロセスがコア・メモリーで動作中であることを示す。
<b>SNOSWAP</b>	0x00000002	プロセスをスワップアウトできないことを示す。
<b>STRC</b>	0x00000008	プロセスがトレースされていることを示す。
<b>SKPROC</b>	0x00000200	カーネル・プロセスを示す。
<b>SEXIT</b>	0x00010000	プロセスが終了していることを示す。
<b>SLPDATA</b>	0x00020000	プロセスで大きなページが使用されることを示す。
<b>SEXECED</b>	0x00200000	プロセスが稼働したことを示す。
<b>SEXECING</b>	0x01000000	プロセスが実行中 (exec を実行中) であることを示す。
<b>SPSEARLYALLOC</b>	0x04000000	このプロセスのページング・スペースが初期に割り当てられることを示す。
<b>TKTHREAD</b>	0x00001000	スレッドがカーネル専用スレッドであることを示す。

注: `/usr/include/sys/proc.h` ファイルの `p_flags` フィールドおよび `/usr/include/sys/thread.h` ファイルの `t_flags` フィールドを調べることによって、すべてのプロセス・フラグおよびスレッド・フラグの定義を参照することができます。

- LIM** (**v** フラグ) `setrlimit` サブルーチンのコールを介して指定され、使用されるメモリーのソフト制限。この制限を指定しなかった場合、`xx` が表示されます。この制限をシステム制限 (すなわち、制限なし) に設定した場合には、値 `UNLIM` が表示されます。
- NI** (**-l** および **l** フラグ) `nice` の値。 `sched other` ポリシーの優先順位の計算に使用されます。
- PID** (すべてのフラグ) プロセスのプロセス ID。
- PGIN** (**v** フラグ) コアにロードされていないページへのプロセスによる参照から生じたディスク入出力回数。
- PPID** (**-f**, **l**, および **-l** フラグ) 親プロセスのプロセス ID。
- PRI** (**-l** および **l** フラグ) プロセスまたはカーネル・スレッドの優先順位。大きい番号ほど優先順位が低い。
- PROJECT**  
(**-P** フラグ) プロセスに割り当てられたプロジェクト名。
- RSS** (**v** フラグ) プロセスの実メモリー (常駐セット) のサイズ (1 キロバイト単位)。
- S** (**-l** および **l** フラグ) プロセスまたはカーネル・スレッドの状態。  
プロセスの場合:
- O** 存在しない
  - A** アクティブ
  - W** スワップ済み
  - I** アイドル (始動待ちの) 状態
  - Z** 取り消し

**T** 停止  
カーネル・スレッドの場合:  
**O** 存在しない  
**R** 実行中  
**S** スリープ中  
**W** スワップ済み  
**Z** 取り消し  
**T** 停止

**SC** (**-o THREAD** フラグ) プロセスまたはカーネル・スレッドの中断カウント。プロセスの中断カウントはカーネル・スレッドの中断カウントの和です。

**SCH** (**-o THREAD** フラグ) カーネル・スレッドのスケジューリング・ポリシー。ポリシー sched\_other、sched\_fifo、および sched\_rr はそれぞれ 0、1、2 と表示されます。

**SIZE** (**v** フラグ) プロセスのデータ・セクションの仮想サイズ (1 キロバイト単位)。

**SHMPGSZ**  
(**Z** フラグ) プロセスの共有メモリー・ページ・サイズ。

**SPGSZ**  
(**Z** フラグ) プロセスのスタック・ページ・サイズ。

**SSIZ** (**s** フラグ) カーネル・スタックのサイズ。マルチスレッド・プロセスの場合、この値は常に 0 (ゼロ) です。

**STAT** (**s**、**u**、および **v** フラグ) プロセスの状態が入っています。

**0** 存在しない  
**A** アクティブ  
**I** 中間  
**Z** 取り消し  
**T** 停止  
**K** 使用可能なカーネル・プロセス

**STIME**  
(**-f** および **u** フラグ) プロセスの始動時刻。**LANG** 環境変数がこのフィールドの表示を制御します。

**SUBPROJ**  
(**-P** フラグ) プロセスに割り当てられたサブプロジェクト ID。

**SZ** (**-l** および **l** フラグ) プロセスのコア・イメージの 1 キロバイト単位のサイズ。

**THCNT**  
(**-o thcount** フラグ) プロセスが所有するカーネル・スレッド数。

**TID** (**-o THREAD** フラグ) カーネル・スレッドのスレッド ID。

**TIME** (すべてのフラグ) プロセスの合計実行時間。時間の表示フォーマットは *mm:ss* または *mmm:ss* (実行時間が 100 分に達した場合) です。これは **-o time** フラグを指定した場合の表示フォーマットとは異なります。

#### **TPGSZ**

(**Z** フラグ) プロセスのテキスト・ページ・サイズ。

**TRS** (**v** フラグ) テキストの常駐セット (実メモリー) のサイズ。

**TSIZ** (**v** フラグ) テキスト (共用プログラム) イメージのサイズ。

**TTY** (すべてのフラグ) プロセスのための制御端末で、以下のとおりです。

- プロセスが端末と関連付けられていない。

? 不明。

*Number*

TTY 番号。例えば、エントリー 2 は TTY2 を示します。

**UID** (**-f**、**-l**、および **l** フラグ) プロセス・オーナーのユーザー ID。 **-f** フラグを指定すると、ログイン名が表示されます。

**USER** (**u** フラグ) プロセス・オーナーのログイン名。

#### **WCHAN**

(**-l** フラグ) プロセスまたはカーネル・スレッドが待機中またはスリープ中であるイベント。カーネル・スレッドの場合、カーネル・スレッドが実行中であれば、このフィールドはブランクになります。プロセスの場合、カーネル・スレッドが 1 つだけスリープしていると、待機チャンネルはスリープ中のカーネル・スレッドの待機チャンネルとして定義されます。それ以外の場合は、アスタリスクが表示されます。

#### **WCHAN**

(**l** フラグ) プロセスが待機中のイベント (システム内のアドレス)。 数値出力が要求されない限り、アドレスをソートする記号が選ばれます。

**%CPU** (**u** および **v** フラグ) プロセス始動以降、プロセスが CPU を使用した時間のパーセンテージ。この値は、プロセスが CPU を使用した時間をプロセスの経過時間で割ることによって算出されます。マルチプロセッサ環境では、同じプロセスにある複数のスレッドは異なる CPU 上で同時に稼働できるため、この値をさらに使用可能な CPU の数で割ります。(このデータを算出するための時間基準は変動するため、すべての **%CPU** フィールドの合計は 100% を超えることがあります。)

#### **%MEM**

(**u** および **v** フラグ) このプロセスによって使用される実メモリーのパーセンテージ。 **%MEM** の値は、他のプロセスとプログラム・テキストを共用するプロセスのコストを誇張する傾向にあります。1 つのプログラムの複数のコピーが実行されていて、プログラム・テキストのコピーがすべてのインスタンスによって共用されている場合、共用による重複は考慮されていません。そのプログラムのインスタンスごとに、テキスト・セクションのサイズが考慮されます。したがって、あるプログラムの複数のコピーが実行されている場合、全プロセスの **%MEM** 値を合計すると 100% を超えることがあります。

既に終了しており、そのプロセスの終了を待っていない親プロセスを持つプロセスは、<defunct> と記されます。終了しようとしてブロックされたプロセスは、<exiting> と記されます。 **ps** コマンドは、メモリーまたはスワップ領域によってプロセスが作成されたときに指定されたファイル名および引数を判別しようとします。

注:

1. **ps** コマンドの実行中に、プロセスが変更することがあります。その場合、defunct プロセスの表示データの一部は無意味なものとなります。
2. **ps** プログラムはメモリーを検査して、プロセスの作成時に使用されたファイル名および引数を検索します。ただし、プロセスが情報を破壊する可能性があるため、このファイル名と引数の検索方法は信頼性を欠きます。

## フラグ

以下のフラグの前には、- (マイナス記号) が付きます。

- A 全てのプロセスに関する情報を標準出力に書き出します。
- a セッション・リーダー、および端末に関連しないプロセスを除くすべてのプロセスに関する情報を標準出力に書き出します。
- c *Clist* *Clist* 変数にリストされたワークロード・マネージメント・クラスに割り当てられたプロセスに関する情報だけを表示します。 *Clist* 変数は、コンマで区切ったクラス名のリストか、二重引用符 (" ") で囲まれたクラス名のリスト (この場合、コンマ、または 1 つ以上のスペース、あるいはこの両方でクラス名同士が区切られています) です。
- d セッション・リーダーを除くすべてのプロセスに関する情報を標準出力に書き出します。
- e カーネル・プロセス以外のすべてのプロセスに関する情報を標準出力に書き出します。
- F *Format* -o *Format* と同じ。
- f 完全なリストを生成します。
- G *Glist* *Glist* 変数にリストされた実効グループのプロセスに関する情報のみを、標準出力に書き出します。 *Glist* 変数は、コンマで区切った実効グループ ID のリストか、または二重引用符 (" ") で囲み、コンマか 1 つ以上のスペースで区切った実効グループ ID のリストです。
- g *Glist* *Glist* 変数にリストされたプロセス・グループのプロセスに関する情報のみを、標準出力に書き出します。 *Glist* 変数は、コンマで区切ったプロセス・グループ ID のリストか、または二重引用符 (" ") で囲み、コンマか 1 つ以上のスペースで区切ったプロセス・グループ ID のリストです。
- k カーネル・プロセスをリストします。
- l 詳細なリストを作成します。 l フラグも参照してください。
- L *pidlist* *pidlist* 変数で PID に渡された、各 PID すべての子孫のリストを生成します。 *pidlist* 変数は、コンマで区切られたプロセス ID のリストです。指定されたすべての PID からの子孫のリストが、プロセス・テーブルに現れた順に印刷されます。
- M すべての 64 ビット・プロセスをリストします。
- m プロセスとともにカーネル・スレッドもリストします。プロセスの出力行に続いて、カーネル・スレッドごとに追加の出力行がリストされます。このフラグは、適切な -o *Format* フラグが指定されない限り、スレッドに固有のフィールド (**bnd**、**scount**、**sched**、**thcount**、および **tid**) を表示しません。
- N スレッド統計情報の収集は行いません。このフラグを指定すると、**ps** は、プロセスのスレッド・チェーンをスキャンしなくても入手できる統計情報だけをレポートします。
- n *NameList* デフォルトの代わりに代替システム名リスト・ファイルを指定します。オペレーティング・システムでは、情報は直接カーネルに提供されるため、-n フラグは使用されません。



*Format* 変数で指定されたフォーマットで情報を表示します。*Format* 変数には、複数のフィールド指定子を指定できます。*Format* 変数は、コンマで区切ったフィールド指定子のリストか、または 1 組の " " (二重引用符) で囲み、コンマか 1 つ以上のスペースで区切ったフィールド指定子のリストか、あるいはその両方です。

それぞれのフィールド指定子にはデフォルトのヘッダーがあります。デフォルトのヘッダーをオーバーライドするには、= (等号) の後にヘッダーのユーザー定義テキストを指定します。フィールドは、コマンド行に指定した順番で欄フォーマットで書き込まれます。フィールド幅は、少なくともデフォルト幅かユーザー定義ヘッダー・テキストの幅になるように、システムによって指定されます。ヘッダー・テキストが null の場合 (例えば、**-o user=** が指定された場合)、フィールド幅は少なくともデフォルトのヘッダー・テキストと同じ幅になります。すべてのヘッダー・フィールドが null の場合、ヘッダー行は書き込まれません。

システムが認識するフィールド指定子を以下に示します。

- args** 実行中のコマンドの完全名を示します。切り捨てが行われることがあります。すべてのコマンド行引数が含まれます。このフィールドのデフォルト・ヘッダーは **COMMAND** です。
- bnd** プロセスまたはカーネル・スレッドがバインドされているプロセッサ (もしあれば) を示します。このフィールドのデフォルト・ヘッダーは **BND** です。
- class** プロセスまたはスレッドに割り当てられたワークロード・マネージメント・クラスを示します。このフィールドのデフォルト・ヘッダーは **CLASS** です。
- comm** 実行中のコマンドの省略名を示します。コマンド行引数は含まれません。このフィールドのデフォルト・ヘッダーは **COMMAND** です。
- cpu** プロセスのスケジューリング優先順位を決定します。プロセスやスレッドの CPU 使用状況は、システム・クロックが時を刻むごとに増加するので、プロセスやスレッドが実行中であることが分かります。スケジューラーは、毎秒一度ずつ 2 で除算してこの値を減分させます。sched\_other ポリシーの場合、大きな値は、CPU 処理中心のプロセスを示し、プロセス優先順位が低くなりますが、小さい値は、入出力処理中心のプロセスを示し、優先順位がより適切になります。
- dpgsz** プロセスのデータ・ページ・サイズを示します。
- etime** プロセス始動後の経過時間を示します。経過時間は以下のフォーマットで表示されます。
- [[ dd-]hh:]mm:ss
- dd* は日数を、*hh* は時間数を、*mm* は分数を、*ss* は秒数を指定します。このフィールドのデフォルト・ヘッダーは **ELAPSED** です。
- group** プロセスの実効グループ ID を示します。テキストによるグループ ID が表示されます。テキストによるグループ ID を獲得できない場合には、10 進表記が使用されます。このフィールドのデフォルト・ヘッダーは **GROUP** です。
- nice** プロセスの nice の値の 10 進値を示します。このフィールドのデフォルト・ヘッダーは **NI** です。

-o Format (続き)

<b>pcpu</b>	使用した CPU 時間と使用可能な CPU 時間の比率を、パーセンテージで示します。このフィールドのデフォルト・ヘッダーは <b>%CPU</b> です。
<b>pgid</b>	プロセスのグループ ID の 10 進値を示します。このフィールドのデフォルト・ヘッダーは <b>PGID</b> です。
<b>pid</b>	プロセス ID の 10 進値を示します。このフィールドのデフォルト・ヘッダーは <b>PID</b> です。
<b>ppid</b>	親プロセス ID の 10 進値を示します。このフィールドのデフォルト・ヘッダーは <b>PPID</b> です。
<b>rgroup</b>	プロセスの実グループ ID を示します。テキストによるグループ ID が表示されます。テキストによるグループ ID を獲得できない場合には、10 進表記が使用されます。このフィールドのデフォルト・ヘッダーは <b>RGROUP</b> です。
<b>ruser</b>	プロセスの実ユーザー ID を示します。テキストによるユーザー ID が表示されます。テキストによるユーザー ID を獲得できない場合には、10 進表記が使用されます。このフィールドのデフォルト・ヘッダーは <b>RUSER</b> です。
<b>scount</b>	カーネル・スレッドの中断カウントを示します。このフィールドのデフォルト・ヘッダーは <b>SC</b> です。
<b>sched</b>	カーネル・スレッドのスケジューリング・ポリシーを示します。このフィールドのデフォルト・ヘッダーは <b>SCH</b> です。
<b>shmpgsz</b>	プロセスの共有メモリー・ページ・サイズを示します。
<b>spgsz</b>	プロセスのスタック・ページ・サイズを示します。
<b>tag</b>	ワークロード・マネージャーのアプリケーション・タグを示します。このフィールドのデフォルト・ヘッダーは <b>TAG</b> です。このタグは、最高 30 文字の長さの文字列で、 <b>ps</b> で表示する場合は切り捨てることができます。タグを設定しないプロセスの場合、このフィールドは - (ハイフン) として表示されます。
<b>tcpu</b>	CPU 時間の合計。これは、単一プロセスの場合の CPU 時間の累計を示します。このフィールドのデフォルト・ヘッダーは <b>TCPU</b> です。
<b>tctime</b>	接続時間の合計。1 つのログイン・セッションがアクティブであることが可能な時間の合計を示します。この値に意味があるのは、セッション・リーダー・プロセスの場合だけです。このフィールドのデフォルト・ヘッダーは <b>TCTIME</b> です。
<b>tdiskio</b>	ディスク入出力の合計。これは、単一プロセスの場合のディスク入出力のブロック数の累計を示します。このフィールドのデフォルト・ヘッダーは <b>TDISKIO</b> です。
<b>tpgsz</b>	プロセスのテキスト・ページ・サイズを示します。
<b>vmsize</b>	WLM 仮想メモリー限度を示します。これを使用すると、新規ヘッダー <b>VMSIZ</b> が表示されます。 <b>VMSIZ</b> は、プロセスによって使用される仮想メモリーを表示します。この値は 1 MB 単位で表されます。

### thcount

プロセスが所有するカーネル・スレッド数を示します。このフィールドのデフォルト・ヘッダーは **THCNT** です。

### THREAD

以下のフィールドを示します。

- ユーザー名 (**uname** フィールド)
- プロセスおよび各プロセスの親プロセス ID (**pid** および **ppid** フィールド)
- スレッドのカーネル・スレッド ID (**tid** フィールド)
- プロセスまたはカーネル・スレッドの状態 (**S** フィールド)
- プロセスまたはカーネル・スレッドの CPU 使用状況 (**C** フィールド)
- プロセスまたはカーネル・スレッドの優先順位 (**PRI** フィールド)
- プロセスまたはカーネル・スレッドの中断カウント (**scount** フィールド)
- プロセスまたはカーネル・スレッドの待機チャンネル (**WCHAN** フィールド)
- プロセスまたはカーネル・スレッドのフラグ (**F** フィールド)
- プロセスの制御端末 (**tty** フィールド)
- プロセスまたはカーネル・スレッドのバインド先である CPU (**bnd** フィールド)
- プロセスによって実行中のコマンド (**comm** フィールド)。

**-o THREAD** フラグを指定しても、**-m** フラグも指定しなければ、スレッドは表示されません。

### tid

カーネル・スレッドのスレッド ID を示します。このフィールドのデフォルト・ヘッダーは **TID** です。

### time

プロセス始動後の CPU の累積時間を示します。時間は以下のフォーマットで表示されます。

[ *dd*-]*hh*:*mm*:*ss*

*dd* は日数を、*hh* は時間数を、*mm* は分数を、*ss* は秒数を指定します。このフィールドのデフォルト・ヘッダーは **TIME** です。

### tty

プロセスの制御端末名を示します。このフィールドのデフォルト・ヘッダーは **TT** です。

### user

プロセスの実効ユーザー ID を示します。テキストによるユーザー ID が表示されます。テキストによるユーザー ID を獲得できない場合には、10 進表記が使用されます。このフィールドのデフォルト・ヘッダーは **USER** です。

### vsz

仮想メモリーでのプロセスのサイズを、キロバイト単位の 10 進整数で示します。このフィールドのデフォルト・ヘッダーは **VSZ** です。

-o *Format* (続き)

上記以外の場合には、フィールド・ディスクリプターも含め、複数フィールドを *Format* 変数による指定フォーマットで表示できます。フィールド・ディスクリプターを *Format* 変数に使用する場合には、そのフィールド・ディスクリプターを二重引用符 (" ") で囲まなければなりません。フィールド・ディスクリプターとフィールド指定子の対応を以下の表に示します。

Field Descriptors	Field Specifiers	Default Headers
%a	args	COMMAND
%c	comm	COMMAND
%t	etime	ELAPSED
%D	dpgsz	DPGSZ
%G	group	GROUP
%n	nice	NI
%C	pcpu	%CPU
%r	pgid	PGID
%p	pid	PID
%P	ppid	PPID
%g	rgroup	RGROUP
%u	ruser	RUSER
%S	spgsz	SPGSZ
%x	time	TIME
%T	tpgsz	TPGSZ gd
%y	tty	TTY
%U	user	USER
%z	vsz	VSZ

それぞれのフィールド指定子にはデフォルトのヘッダーがあります。デフォルトのヘッダーをオーバーライドするには、等号 (=) の後にヘッダーのユーザー定義テキストを指定します。フィールドは、コマンド行に指定した順番で欄フォーマットで書き込まれます。フィールド幅は、少なくともデフォルト幅かユーザー定義ヘッダー・テキストの幅になるように、システムによって指定されます。ヘッダー・テキストが `null` の場合 (例えば、`-o user=` が指定された場合)、フィールド幅は少なくともデフォルトのヘッダー・テキストと同じ幅になります。すべてのヘッダー・フィールドが `null` の場合、ヘッダー行は書き込まれません。

**-o** *Format* (続き)

以下は、デフォルト・ヘッダーとさまざまなフィールド指定子の間のマッピングです。「Default Header」列の各エントリーは、「Field specifier」の対応するエントリーに等号 (=) を追加し、その後にヘッダーのユーザー定義テキストを指定することによってオーバーライドすることができます。

Default Header	Field specifier
ARGS	"args"
COMM	"comm"
COMM	"command"
COMM	"ucomm"
F_ETIME	"etime"
GROUP	"group"
GROUP	"gname"
GID	"gid"
NICE	"nice"
PRI	"pri"
NICE	"ni"
PCPU	"pcpu"
PMEM	"pmem"
PGID	"pgid"
PID	"pid"
PPID	"ppid"
RGROUP	"rgroup"
RGROUP	"rgname"
RGID	"rgid"
RUSER	"ruser"
RUSER	"runame"
RUID	"ruid"
TIME	"time"
TIME	"cputime"
TTY	"tty"
TTY	"tt"
TTY	"tname"
TTY	"longtname"
USER	"user"
USER	"uname"
UID	"uid"
LOGNAME	"logname"
STIME	"start"
VSZ	"vsz"
VSZ	"vsize"
RSS	"rssize"
FLAG	"flag"
STATUS	"status"
CP	"cp"
PAGEIN	"pagein"
WCHAN	"wchan"
NWCHAN	"nwchan"
ST	"st"
TID	"tid"
SCOUNT	"scount"
BIND	"bnd"
SCHED	"sched"
THCOUNT	"thcount"
TAG	"tag"
CLASS	"class"
TCPU	"tcpu"
TDISKIO	"tdiskio"
TCTIME	"tctime"
MACLAB	"mac"

**-p** *Plist*

*Plist* 変数に指定されたプロセス番号を持つプロセスに関する情報のみを表示します。*Plist* 変数は、コンマで区切ったプロセス ID 番号のリストか、または二重引用符 (" ") で囲み、コンマか 1 つ以上のスペースで区切ったプロセス ID 番号のリスト、あるいはその両方です。

- P** プロジェクトのプロジェクト名、プロジェクト・オリジン (Project origin)、およびサブプロジェクト ID を表示します。スティッキー・ビットがプロセスに設定されている場合、プロジェクト名の前にアスタリスク (\*) 文字が付きます。 **Project origin** フィールドにより、現在ロードされているプロジェクト・リポジトリ (LOCAL または LDAP) を指定します。
- t Tlist** *Tlist* 変数にリストされた制御 tty に関連したプロセスに関する情報のみを表示します。*Tlist* 変数は、コンマで区切った tty ID のリストか、二重引用符 ( " ") で囲み、それぞれをコンマまたは 1 つ以上のスペースで区切った tty ID のリストのいずれか、あるいはその両方です。
- T pid** 指定された PID を root とするプロセス階層を、ASCII アートを使用してツリー・フォーマットで表示します。このフラグは、**-f** フラグ、**-F** フラグ、**-o** フラグ、および **-l** フラグと組み合わせて使用できます。
- U Ulist** *Ulist* 変数に指定されたユーザー ID 番号またはログイン名を持つプロセスに関する情報のみを表示します。*Ulist* 変数は、コンマで区切ったユーザー ID のリストか、または二重引用符 ( " ") で囲み、コンマか 1 つ以上のスペースで区切ったユーザー ID のリストです。リストでは、**ps** コマンドは、**-f** フラグが指定されない限り数値のユーザー ID を表示し、その後でコマンドがログイン名を表示します。このフラグは **-u Ulist** フラグと同等です。**u** フラグも参照してください。
- u Ulist** このフラグは **-U Ulist** フラグと同等です。
- X** 各ユーザー名/グループ名の使用可能なすべての文字を、最初の 8 文字で切り捨てずに印刷します。
- Z** プロセスのページ・サイズ設定を表示します。

#### DPGSZ

プロセスのデータ・ページ・サイズを示します。

#### SHMPGSZ

プロセスが割り当てる共有メモリー・ページ・サイズを示します。

#### SPGSZ

プロセスのスタック・ページ・サイズを示します。

**TPGSZ** プロセスのテキスト・ページ・サイズを示します。

## オプション

以下のオプションの前には負符号 (-) が付きません。

- a** 端末を使用するすべてのプロセスに関する情報を表示します (通常はユーザー自身のプロセスのみが表示されます)。
- c** プロセス・アドレス・スペースに格納されたコマンド・パラメーターではなく、アカウント用システムに格納されたコマンド名を表示します。
- e** 80 文字までの制限付きで、コマンドのパラメーターとともに環境変数を表示します。
- ew** **e** フラグからの表示を、1 行余分にラップします。
- eww** **e** フラグからの表示を、必要な回数だけラップします。
- ewww** **e** フラグからの表示をラップし、**INT\_MAX** 値に到達するまで **ENV** リストを表示します。
- g** すべてのプロセスを表示します。
- l** **F**、**S**、**UID**、**PID**、**PPID**、**C**、**PRI**、**NI**、**ADDR**、**SZ**、**PSS**、**WCHAN**、**TTY**、**TIME**、および **CMD** フィールドを持つ詳細リストを表示します。
- n** 数値出力を表示します。詳細なリストでは、**WCHAN** フィールドは記号ではなく数値で表示されます。ユーザーのリストでは、**USER** フィールドは **UID** フィールドに置き換えられます。
- s** (システム保守担当者によって使用される) 各プロセスのカーネル・スタックのサイズ (SSIZ) を、基本出力フォーマットで表示します。マルチスレッド・プロセスの場合、この値は常に 0 (ゼロ) です。

- t** *tty* 制御している *tty* が *tty* 変数の値であるプロセスを表示します。Tty 変数は、**ps** コマンドが表示するとおりに指定してください。つまり、端末 **/dev/tty/0** の場合には **0**、端末 **/dev/lft0** の場合には **1ft0**、端末 **/dev/pts/2** の場合には **pts/2** と指定してください。
- u** ユーザー向け出力を表示します。これには、**USER**、**PID**、**%CPU**、**%MEM**、**SZ**、**RSS**、**TTY**、**STAT**、**STIME**、**TIME**、および **COMMAND** フィールドが含まれます。
- v** **PGIN**、**SIZE**、**RSS**、**LIM**、**TSIZ**、**TRS**、**%CPU**、**%MEM** フィールドを表示します。
- w** 幅の広いフォーマットで出力します (80 ではなく 132 桁)。繰り返すと、(例えば、**ww**) 任意に幅の広い出力が使用されます。この情報を使って、コマンドをどれくらいの長さで表示するかが決められます。
- x** 制御する端末を持つプロセスに加えて、制御する端末を持たないプロセスも表示します。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0** 正常終了。
- >0** エラーが発生しました。

## 例

1. すべてのプロセスを表示するには、以下のように入力します。

```
ps -e -f
```

すべてのプロセスを、展開されたユーザー名を使用して表示するには、以下のように入力します。

```
ps -X -e -f
```

2. 特定のユーザーが所有するプロセスをリストするには、以下のように入力します。

```
ps -f -l -ujim,jane,su
```

3. **/dev/console** および **/dev/tty1** の *tty* に関連するプロセスをリストするには、以下のように入力します。

```
ps -t console,tty/1
```

4. 端末と関連していないプロセスをリストするには、以下のように入力します。

```
ps -t -
```

5. 指定フォーマットをフィールド指定子付きで表示するには、以下のように入力します。

```
ps -o ruser,pid,ppid=parent,args
```

出力は以下のようになります。

```
RUSER  PID    parent  COMMAND
helene  34     12      ps -o ruser,pid,ppid=parent,args
```

6. 指定フォーマットをフィールド・ディスクリプター付きで表示するには、以下のように入力します。

```
ps -o "< %u > %p %y : %a"
```

出力は以下のようになります。

```
< RUSER >      PID    TT :    COMMAND
< helene >      34     pts/3 : ps -o < %u > %p %y : %a
```

7. 現在の端末によって制御されるプロセスおよびカーネル・スレッドに関する情報を表示するには、以下のように入力します。

```
ps -lm
```

出力は以下のようになります。



```

      F S UID  PID PPID  C PRI NI ADDR  SZ WCHAN  TTY  TIME  CMD
240003 A  26 8984 7190  1  60 20 2974 312          pts/1  0:00 -ksh
      400 S  -  -  -  1  60 -  -  -          -  -  -
200005 A  26 9256 8984 15  67 20 18ed 164          pts/1  0:00 ps
      0 R  -  -  - 15  67 -  -  -          -  -  -

```

8. すべてのプロセスおよびカーネル・スレッドに関する情報を表示するには、以下のように入力します。

```
ps -emo THREAD
```

出力は以下のようになります。

```

USER  PID  PPID  TID S  C  PRI SC  WCHAN  FLAG  TTY  BND  CMD
jane  1716 19292  - A 10  60  1          * 260801 pts/7  -  biod
-    -    - 4863 S  0  60  0 599e9d8 8400  -  -  -
-    -    - 5537 R 10  60  1 5999e18 2420  -  3  -
luke  19292 18524  - A  0  60  0 586ad84 200001 pts/7  -  -ksh
-    -    - 7617 S  0  60  0 586ad84 400  -  -  -
luke  25864 31168  - A 11  65  0          - 200001 pts/7  -  -
-    -    - 8993 R 11  65  0          - 0  -  -  -

```

9. すべての 64 ビット・プロセスをリストするには、次のように入力します。

```
ps -M
```

10. プロセスのプロジェクト割り当ての詳細を表示するには、次のように入力します。

```
ps -P
```

11. プロセスのページ・サイズ設定を表示するには、次のように入力します。

```
ps -Z
```

出力は以下のようになります。

```

      PID  TTY  TIME  DPGSZ  SPGSZ  TPGSZ  CMD
41856 pts/15 0:00  4K   4K   4K ps
84516 pts/15 0:00  4K   4K   4K ksh

```

## ファイル

<code>/usr/bin/ps</code>	<code>ps</code> コマンドが入っています。
<code>/etc/passwd</code>	ユーザー ID 情報を指定します。
<code>/dev/pty*</code>	端末 (PTY) 名を示します。
<code>/dev/tty*</code>	端末 (TTY) 名を示します。

## 関連情報

`kill` コマンド、`nice` コマンド。

パフォーマンス・マネージメントの `Using the ps command` のセクション。

## System V ps コマンド

### 構文

```

/usr/sysv/bin/ps [ -a ] [ -A ] [ -c ] [ -d ] [ -e ] [ -f ] [ -j ] [ -l ] [ -L ] [ -P ] [ -y ] [ -g pgrplist ]
[ -o format ] [ -p proclist ] [ -s sidlist ] [ -t termlist ] [ { -u | -U } uidlist ] [ -G grplist ] [ -X ]

```

## 説明

**ps** コマンドは、アクティブなプロセスについての情報を印刷します。フラグを指定しないと、**ps** は、制御している端末に関連付けられたプロセスの情報を印刷します。出力には、プロセス ID、端末 ID、累積実行時間、およびコマンド名が含まれます。フラグを使用して表示される情報は、フラグにより異なります。

## 出力

**ps** コマンドで使用されるフラグにより、表示される情報によって列見出しは変化します。各見出しの定義を以下にリストします (これらの見出しを表示させるフラグを括弧内に示します)。

**F (-l)** プロセス (または、**-L** オプションを指定した場合はスレッド) に関連付けられたフラグ (16 進数で追加)。プロセスおよびスレッドに関連した、より重要な **F** フィールド・フラグ (16 進数および追加情報) の一部を以下に示します。

F フィールドの表

フラグ	16 進数	定義
<b>SLOAD</b>	0x00000001	プロセスがコア・メモリーで動作中であることを示す。
<b>SNOSWAP</b>	0x00000002	プロセスをスワップアウトできないことを示す。
<b>STRC</b>	0x00000008	プロセスがトレースされていることを示す。
<b>SKPROC</b>	0x00000200	カーネル・プロセスを示す。
<b>SEXIT</b>	0x00010000	プロセスが終了していることを示す。
<b>SEXECED</b>	0x00200000	プロセスが稼働したことを示す。
<b>SEXECING</b>	0x01000000	プロセスが実行中 (exec を実行中) であることを示す。
<b>TKTHREAD</b>	0x00001000	スレッドがカーネル専用スレッドであることを示す。

注: **/usr/include/sys/proc.h** ファイルの **p\_flags** フィールドおよび **/usr/include/sys/thread.h** ファイルの **t\_flags** フィールドを調べることによって、すべてのプロセス・フラグおよびスレッド・フラグの定義を見ることができます。

**S (-l)** プロセスまたはカーネル・スレッドの状態:

プロセスの場合:

- O** 存在しない
- A** アクティブ
- W** スワップ済み
- I** アイドル
- Z** 取り消し
- T** 停止

カーネル・スレッドの場合:

- O** 存在しない
- R** 実行中
- S** スリープ中

**W**     スワップ済み  
**Z**     取り消し  
**T**     停止

**UID (-f, -l)**

プロセスのユーザー ID 番号 (-f オプションを指定するとログイン名が表示される)。

**PID (all)**

プロセスのプロセス ID。

**PPID (-f, -l)**

親プロセスのプロセス ID。

**CLS (-c)**

プロセスのスケジューリング・クラス。-c フラグを使用した場合のみ印刷されます。

**NI (-l) sched\_other** ポリシーの優先順位の計算に使用される、プロセスの nice の値。

**PRI (-c, -l)**

プロセスまたはカーネル・スレッドの優先順位。高い番号は低い優先順位を意味します。

**ADDR (-l)**

通常の場合は、プロセス・スタックのセグメント番号が入っています。カーネル・プロセスの場合は、プリプロセス・データ領域のアドレスです。

**SZ (-l)**

プロセスのコア・イメージのサイズ (ページ数)。

**WCHAN(-l)**

プロセスまたはカーネル・スレッドが待機中またはスリープ中であるイベント。カーネル・スレッドの場合、カーネル・スレッドが実行中であれば、このフィールドはブランクになります。プロセスの場合、カーネル・スレッドが 1 つだけスリープしていると、待機チャンネルはスリープ中のカーネル・スレッドの待機チャンネルとして定義されます。それ以外の場合は、アスタリスクが表示されます。

**STIME (-f, -u)**

プロセスの始動時刻。LANG 環境変数がこのフィールドの表示を制御します。

**TTY (all)**

プロセスのための制御端末で、以下のとおりです。

-       プロセスが端末と関連付けられていない。  
?       不明

**TIME (all)**

プロセスの合計実行時間。時間の表示フォーマットは mm:ss または mmmm:ss (実行時間が 100 分に達した場合) です。これは -o time フラグを指定した場合の表示フォーマットとは異なります。

**LTIME (-L)**

個々の LWP の実行時間。

**CMD (all)**

コマンド名を含む。完全なコマンド名およびそのパラメーターは、-f フラグで表示されます。

**LWP (-L)**

カーネル・スレッドの tid。

## NLWP(-Lf)

プロセス内のカーネル・スレッドの数。

## PSR (-P)

カーネル・スレッドがバインドされる論理プロセッサ番号 (バインドされる場合)。プロセスの場合は、そのすべてのスレッドが同じプロセッサにバインドされたときに、このフィールドが表示されます。

## RSS (-ly)

プロセスの実メモリ (常駐セット) サイズ (1 KB 単位)。

## フォーマット

次のリストは、システムにより認識されるフィールド指定子を説明しています。これらのフィールド指定子は、**-o** フラグと一緒に使用することができ、**ps** コマンドの出力フォーマットを指定します。

システムが認識するフィールド指定子を以下に示します。

**addr** 通常は、プロセス・スタックのセグメント番号を示します。カーネル・プロセスの場合は、プリプロセス・データ領域のアドレスです。このフィールドのデフォルト・ヘッダーは **ADDR** です。

**args** 実行中のコマンドの完全名を示します。切り捨てが行われることがあります。すべてのコマンド行引数が含まれます。このフィールドのデフォルト・ヘッダーは **COMMAND** です。

**c** プロセスやスレッドの CPU 使用状況は、システム・クロックが時を刻むごとに増加するので、プロセスやスレッドが実行中であることが分かります。スケジューラーは、毎秒一度ずつ 2 で除算してこの値を減分させます。**sched\_other** ポリシーでは、CPU 使用率はプロセスのスケジューリング優先順位を決定する際に使用されます。大きな値は CPU 処理中心のプロセスを示し、プロセス優先順位が低くなりますが、小さな値は入出力処理中心のプロセスを示し、優先順位がそれより高くなります。このフィールドのデフォルト・ヘッダーは **C** です。

**class** カーネル・スレッドのスケジューリング・ポリシーを示します。ポリシーは、**sched\_other**、**sched\_fifo** および **sched\_rr** です。このフィールドのデフォルト・ヘッダーは **CLS** です。

**comm** 実行中のコマンドの省略名を示します。コマンド行引数は含まれません。このフィールドのデフォルト・ヘッダーは **COMMAND** です。

**etime** プロセス始動後の経過時間を示します。経過時間は以下のフォーマットで表示されます。

```
[[ dd -] hh: ]mm :ss
```

*dd* は日数を、*hh* は時間数を、*mm* は分数を、*ss* は秒数を指定します。

このフィールドのデフォルト・ヘッダーは **ELAPSED** です。

**f** プロセスに関連付けられたフラグ (16 進数で追加) を示します。このフィールドのデフォルト・ヘッダーは **COMMAND** です。

**fname** プロセスの実行可能ファイルのベース名の最初の 8 バイトを示します。このフィールドのデフォルト・ヘッダーは **COMMAND** です。

**gid** プロセスの実効グループ ID 番号を、10 進の整数として示します。このフィールドのデフォルト・ヘッダーは **GID** です。**-f** フラグを指定すると、ログイン名が出力されます。

**group** プロセスの実効グループ ID を示します。テキストによるグループ ID が表示されます。テキストによるグループ ID を獲得できない場合には、10 進表記が使用されます。このフィールドのデフォルト・ヘッダーは **GROUP** です。

- lwp** カーネル・スレッドのスレッド ID を示します。このフィールドのデフォルト・ヘッダーは **TID** です。
- nice** プロセスの nice の値の 10 進値を示します。このフィールドのデフォルト・ヘッダーは **NI** です。
- nlwp** プロセスが所有するカーネル・スレッド数を示します。このフィールドのデフォルト・ヘッダーは **THCNT** です。
- pcpu** 使用した CPU 時間と使用可能な CPU 時間の比率を、パーセンテージで示します。このフィールドのデフォルト・ヘッダーは **%CPU** です。
- pgid** プロセスのグループ ID の 10 進値を示します。このフィールドのデフォルト・ヘッダーは **PGID** です。
- pid** プロセス ID の 10 進値を示します。このフィールドのデフォルト・ヘッダーは **PID** です。
- pmem** このプロセスで使用される実メモリーのパーセントを示します。このフィールドのデフォルト・ヘッダーは **%MEM** です。
- ppid** 親プロセス ID の 10 進値を示します。このフィールドのデフォルト・ヘッダーは **PPID** です。
- pri** プロセスまたはカーネル・スレッドの優先順位を示し、高い番号は低い優先順位を意味します。このフィールドのデフォルト・ヘッダーは **PRI** です。
- psr** カーネル・スレッドがバインドされるプロセッサの論理プロセッサ番号を示します (ある場合)。このフィールドのデフォルト・ヘッダーは **PSR** です。
- rgid** プロセスの実グループ ID 番号を、10 進の整数として示します。このフィールドのデフォルト・ヘッダーは **RGID** です。
- rgroup** プロセスの実グループ ID を示します。テキストによるグループ ID が表示されます。テキストによるグループ ID を獲得できない場合には、10 進表記が使用されます。このフィールドのデフォルト・ヘッダーは **RGROUP** です。
- rss** プロセスの実メモリー (常駐セット) サイズを示します (1 KB 単位で)。このフィールドのデフォルト・ヘッダーは **RSS** です。
- ruid** プロセスの実ユーザー ID 番号を、10 進の整数として示します。このフィールドのデフォルト・ヘッダーは **RUID** です。
- ruser** プロセスの実ユーザー ID を示します。テキストによるユーザー ID が表示されます。テキストによるユーザー ID を獲得できない場合には、10 進表記が使用されます。このフィールドのデフォルト・ヘッダーは **RUSER** です。
- s** プロセスの状態を示します。このフィールドのデフォルト・ヘッダーは **S** です。
- sid** セッション・リーダーのプロセス ID を示します。このフィールドのデフォルト・ヘッダーは **SID** です。
- stime** プロセスの開始時刻を示します。LANG 環境変数がこのフィールドの表示を制御します。このフィールドのデフォルト・ヘッダーは **STIME** です。
- time** プロセス始動後の CPU の累積時間を示します。時間は **etime** と同じフォーマットで表示されます。このフィールドのデフォルト・ヘッダーは **TIME** です。
- tty** プロセスの制御端末名を示します。このフィールドのデフォルト・ヘッダーは **TT** です。
- uid** プロセスの実効ユーザー ID 番号を 10 進の整数として示します。このフィールドのデフォルト・ヘッダーは **UID** です。

**user** プロセスの実効ユーザー ID を示します。テキストによるユーザー ID が表示されます。テキストによるユーザー ID を獲得できない場合には、10 進表記が使用されます。このフィールドのデフォルト・ヘッダーは **USER** です。

**vsz** プロセスのコア・イメージのサイズを、キロバイト単位の 10 進整数で示します。このフィールドのデフォルト・ヘッダーは **VSZ** です。

#### wchan

プロセスまたはカーネル・スレッドが待機中またはスリープ中であるイベントを示します。カーネル・スレッドの場合、カーネル・スレッドが実行中であれば、このフィールドは空白になります。プロセスの場合、カーネル・スレッドが 1 つだけスリープしていると、待機チャンネルはスリープ中のカーネル・スレッドの待機チャンネルとして定義されます。それ以外の場合は、アスタリスクが表示されます。

このフィールドのデフォルト・ヘッダーは **WCHAN** です。

## フラグ

いくつかのフラグは引数としてリストを受け付けます。リスト内の項目は、コンマで区切るかまたは、二重引用符で囲んでコンマまたはスペースで区切ります。 *proclist* と *pgrplist* の値は数字でなければなりません。

- a** セッション・リーダー、および端末に関連しないプロセスを除くすべてのプロセスに関する情報を標準出力に書き出します。
- A** すべてのプロセスに関する情報を標準出力に書き出します。
- c** スケジューラ属性を反映したフォーマットで情報を印刷します。 **-c** フラグは、以下に記述されている、**-f** および **-l** フラグの出力に影響します。
- d** セッション・リーダーを除くすべてのプロセスに関する情報を、標準出力に書き込みます。
- e** カーネル・プロセスを除くすべてのプロセスに関する情報を、標準出力に書き込みます。
- f** 完全なリストを生成します。
- g *pgrplist*** *pgrplist* に指定されたプロセス・グループにあるプロセスに関する情報のみを、標準出力に書き込みます。 *pgrplist* の値は数字でなければなりません。
- G *grplist*** *grplist* に指定されたプロセス・グループにあるプロセスに関する情報のみを、標準出力に書き込みます。 **-G** フラグはグループ名を受け付けます。
- j** セッション ID とプロセス・グループ ID を表示します。
- l** 詳細なリストを作成します。
- L** プロセス内のアクティブ・スレッドの状況を印刷します。
- o *format*** *format* で指定されたフォーマットで情報を表示します。 *format* 変数には、複数のフィールド指定子を指定できます。 **-o** フラグと一緒に使用できるフィールド指定子は、上のフォーマットのセクションに記述されています。
- p *proclist*** *proclist* に指定されたプロセス番号を持つプロセスに関する情報のみを表示します。 *proclist* の値は数字でなければなりません。
- P** プロセスの 1 次カーネル・スレッドがバインドされているプロセッサの、論理プロセッサ番号を表示します (ある場合)。
- s *sidlist*** *sidlist* で指定されたセッション・リーダー ID を持つ、すべてのプロセスを表示します。
- t *termlist*** *termlist* に指定された端末に関連付けられたプロセスに関する情報のみを表示します。
- u *uidlist*** *uidlist* に指定されたユーザー ID 番号またはログイン名を持つプロセスに関する情報のみを表示します。
- U *uidlist*** *uidlist* に指定されたユーザー ID 番号またはログイン名を持つプロセスに関する情報のみを表示します。
- X** フル・ユーザー名またはグループ名を出力します。名前は切り捨てられません。

- y                                    -l オプションと組み合わせると、詳細リストを変更し、“RSS” および “SZ” フィールドをキロバイトで印刷し、“F” および “ADDR” フィールドを印刷しないようにします。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0     正常終了。  
>0   エラーが発生しました。

## 例

- すべてのプロセスを表示するには、以下のように入力します。  
`ps -e -f`
- ユーザー 'guest' が所有するプロセスをリストするには、次のように入力します。  
`ps -f -l -u guest`
- `/dev/pts/0` および `/dev/pts/1` 端末に関連したプロセスをリストするには、次のように入力します。  
`ps -t pts/0,pts/1`
- 端末と関連していないプロセスをリストするには、以下のように入力します。  
`ps -t -`
- 指定フォーマットをフィールド指定子付きで表示するには、以下のように入力します。  
`ps -o ruser,pid,ppid,args`
- プロセス内のすべてのカーネル・スレッドについての情報を表示するには、次のように入力します。  
`ps -L`
- すべてのプロセスのセッション ID とプロセス・グループ ID を表示するには、次のように入力します。  
`ps -jA`
- プロセスのスケジューリング・クラスと優先順位を表示するには、次のように入力します。  
`ps -c -l`
- カーネル・スレッドについての情報および、プロセス内のカーネル・スレッドの数を表示するには、次のように入力します。  
`ps -L -f`
- プロセスまたはカーネル・スレッドがバインドされているプロセッサを表示するには、次のように入力します。  
`ps -P`
- 指定されたプロセス (次の例では `inetd`) の ASCII アートを印刷するには、次のように入力します。  
`ps -T 14220`

出力は次のようになります。

```
  PID   TTY  TIME CMD
 14220   -   0:00 inetd
 16948   -   0:00 | \--telnetd
 32542 pts/4 0:00 |   \--ksh
 26504   -   0:00 | \--telnetd
 41272 pts/5 0:00 |   \--ksh
```



```

26908 pts/5 0:00 | \--vi
28602 - 0:00 | \--telnetd
24830 pts/0 0:00 | \--ksh
676416 pts/0 0:00 | \--ksh
29984 - 0:00 | \--telnetd
38546 pts/6 0:00 | \--ksh
32126 - 0:00 | \--telnetd
11162 pts/7 0:00 | \--ksh
34466 - 0:00 | \--rpc.ttdbserver
35750 - 0:00 | \--telnetd
23612 pts/3 0:00 | \--ksh
36294 - 0:00 | \--telnetd
38096 pts/8 0:00 | \--ksh
39740 - 0:00 | \--telnetd
42226 pts/9 0:01 | \--ksh
40632 - 0:00 | \--telnetd
40232 pts/2 0:00 | \--ksh
32910 pts/2 0:00 | \--dbx
987990 pts/2 0:00 | \--a.out
40722 - 0:00 | \--telnetd
16792 pts/10 0:00 | \--ksh
37886 pts/10 0:00 | \--ps
105716 - 0:00 | \--telnetd
29508 pts/1 0:00 | \--ksh
39478 pts/1 0:00 | \--ksh
38392 pts/1 0:00 | \--vi

```

12. 指定された PID を root とするすべてのプロセスに関する情報を印刷するには、次のように入力します。

```
ps -fL 14220
```

出力は次のようになります。

```

UID  PID  PPID  C   STIME   TTY   TIME CMD
root 14220 8676  0   Apr 07   -    0:00 /usr/sbin/inetd
root 16948 14220  0   Apr 06   -    0:00 telnetd -a
root 23612 35750  0   Apr 10 pts/3   0:00 -ksh
root 24830 28602  1 18:30:56 pts/0   0:00 -ksh
root 28602 14220  0 18:30:55 -    0:00 telnetd -a
root 32542 16948  0   Apr 06 pts/4   0:00 -ksh
root 34466 14220  0   Apr 10   -    0:00 rpc.ttdbserver 100083 1
root 35750 14220  0   Apr 10   -    0:00 telnetd -a
root 40228 24830  8 18:36:01 pts/0   0:00 ps -fL 14220

```

13. すべてのプロセスを、展開されたユーザー名を使用して表示するには、以下のように入力します。

```
ps -X -e -f
```

## ファイル

**/usr/sysv/bin/ps** System V R4 **ps** コマンドが入っています。  
**/etc/passwd** ユーザー ID 情報が入っています。  
**/dev/pty\*** 端末 (PTY) 名を示します。  
**/dev/tty\*** 端末 (TTY) 名を示します。

## 関連情報

**kill** コマンド、 **nice** コマンド。

---

## ps4014 コマンド

### 目的

Tektronix 4014 ファイルを PostScript フォーマットに変換します。

### 構文

```
ps4014 [ -m ] [ -C ] [ -N ] [ -R ] [ -sWidth,Height ] [ -lLeft,Bottom ] [ -SWidth ] [ -pOutFile ] [ File ]
```

### 説明

**ps4014** コマンドは、Tektronix 4014 フォーマット・ファイルで読み取り、PostScript プリンターで印刷するためにそれを PostScript フォーマットに変換します。ファイルを指定しないと、標準入力を使用されます。変換された PostScript ファイルは、標準出力または名前付きファイルに送信することができます。

注: デフォルトでは、4014 のイメージの大きさは、横方向のほぼページ全体になっています。

### フラグ

注: **-m**、**-C**、および **-N** フラグは、4014 コマンドの解釈に影響する 4014 ハードウェア・オプションに値を指定します。

<b>-C</b>	左マージンのペン位置に復帰を移動させますが、次行の下には移動させません。デフォルトでは、復帰コマンドは、次行の下と左マージンの上にペンを移動します。
<b>-lLeft,Bottom</b>	変換済みラスタ・イメージの左下隅の印刷ページ上のロケーションを指定します。 <i>Left</i> および <i>Bottom</i> パラメーターで指定した値は、印刷ページの左下隅からイメージの左下隅までの距離 (インチ単位) です。
<b>-m</b>	4014 の ``Margin 2'' モードを使用可能にします。
<b>-N</b>	次行の下のペン位置に行送りを移動させますが、左マージンには移動させません。デフォルトでは、行送りは、次行の下と左マージンの上にペンを移動します。
<b>-pOutFile</b>	標準出力にではなくて <i>OutFile</i> パラメーターで指定したファイルに、PostScript ファイルが書き込まれるようにします。
<b>-R</b>	縦方向のページ上でイメージを 90 度回転します。デフォルトは、横方向です。
<b>-sWidth,Height</b>	印刷ページの変換済みラスタ・イメージのサイズを指定します。 <i>Width</i> および <i>Height</i> パラメーターは、印刷ページの結果イメージの寸法 (インチ単位) を指定します。
<b>-SWidth</b>	形状を壊さずにイメージの大きさを測れるようにします。 <i>Width</i> パラメーターは、印刷ページの結果イメージの幅 (インチ単位) を指定します。イメージの縦の長さは、出力イメージの縦と横の比率が入カラスタ・フォーマット・ファイルの場合と同じになるように計算されます。

### 各国文字サポート

文字変換についての詳細は、NLSvec ファイルのセクションを参照してください。

### 環境変数

**PSLIBDIR** **ps4014** プロローグの */usr/lib/ps* の代わりに使用されるディレクトリーのパス名。

# ファイル

/usr/lib/ps/ps4014.pro

デフォルトのプロローグ・ファイルが入っています。

## 関連情報

NLSvec ファイル。

---

## ps630 コマンド

### 目的

Diablo 630 印刷ファイルを PostScript フォーマットに変換します。

### 構文

**ps630** [ **-fBodyfont** ] [ **-pFile** ] [ **-sPitch** ] [ **-FBoldfont** ] [ *File...* ]

### 説明

**ps630** コマンドは、Diablo 630 印刷ファイルを PostScript プリンターで印刷できるように PostScript フォーマットに変換します。 *File* 変数を指定しないと、**ps630** コマンドは標準入力から読み取ります。デフォルトでは、PostScript ファイルが標準出力へ送信されます。

**ps630** コマンドは、**-Txerox** フラグで作成した **nroff** ファイルを変換することができます。活字ホイールのエミュレーション情報は、オプションとして指定できます。(太字と正規の) フォント指定は、PostScript フォント名 (Times-Roman、Times-Bold、Courier-Bold、Courier-BoldOblique など) です。1 インチ当たり 10、12、または 15 文字を選択することができます。

アプリケーションによっては、文字を 2 回打つことによって太字を作るものもあります。このタイプの太字は、PostScript フォーマットに変換されません。適切な Diablo コマンド・シーケンス (Esc-0) を打つことによって生まれた太字効果だけが、太字文字を作ります。

**ps630** コマンドの出力では、ページを逆にすることができません。アプリケーションによってページ幅の調整が行われていない場合には、Times-Roman と Helvetica は狭いフォントなので、詰められて見える可能性があります。

以下の Diablo 630 コマンドは、サポートされません。

- 印刷抑制
- HY- プロット
- 拡張文字セット
- プリント・ホイール情報またはプログラム・モードのダウンロード
- 11 インチ以外のページ長さ
- 紙送り制御
- ハンマー・エネルギー制御
- リモート診断
- 逆方向印刷制御

**注:** 反転印刷のための Diablo 630 コマンドがサポートされます。

## フラグ

- fBodyfont** 通常の印刷に使用されるフォントを設定します。デフォルトは Courier です。
- pFile** 標準出力にではなくて *File* パラメーターで指定されたファイルに、PostScript ファイルが書き込まれるようにします。
- sPitch** 印刷用タイプのサイズを選択します (太字と正規の両フォントがこのサイズにスケールされます)。ピッチは 1 インチ当たりの文字数であり、10、12、または 15 のいずれかとします。デフォルトは 12 です。
- FBoldfont** 太字タイプに使用されるフォントを設定します。デフォルトは Courier-Bold です。

## 各国文字サポート

文字変換についての詳細は、NLSvec ファイルのセクションを参照してください。

## 環境変数

**PSLIBDIR** **ps630** プロローグ用 **/usr/lib/ps** ディレクトリーの代わりに使用されるディレクトリーのパス名。

## ファイル

**/usr/lib/ps** PostScript フォントが入っています。

## 関連情報

**enscript** コマンド、**nroff** コマンド。

**NLSvec** ファイル。

---

## psc または psdit コマンド

### 目的

**troff** の中間フォーマットを PostScript フォーマットに変換します。

### 構文

```
{ psc | psdit } [ -f1 CodeSet:Font ] [ -FFontDirectory ] [ -MMediaName ] [ -pPrologue ]  
[ -oList ] [ File ]
```

### 説明

**psc** コマンドと **psdit** コマンドは、デバイス独立の **troff** で作成されたファイルを PostScript プリンターで印刷できるように PostScript フォーマットに変換します。ファイルを指定しないと、標準入力を使用されます。PostScript ファイルが標準出力へ送信されます。

注: **psc** と **psdit** コマンドの入力は、**troff** や **pic** コマンドのような対応する **-Tpsc** オプションで準備すべきです。

**psc** と **psdit** コマンドは、フォント・ファイル (**/usr/lib/font/devpsrc/R**) のプリンターのコード・フィールドを変更することによって作成された拡張文字を扱うことができます。変更されたフィールドには、二重

引用符で囲まれた文字列が含まれます。文字列に含まれるのは、**\b** (バックスラッシュ、b) と、それに続いて、重ね打ちによって新しい文字に構成される標準フォントからの文字シーケンスです。

**psc** コマンドと **psdit** コマンドを使用すれば、**troff** コマンドで、作成された PostScript ファイルに任意の PostScript コードを含めるようになります。**psc** と **psdit** コマンドは、**troff** 中間ファイル・フォーマットで未定義の **%** (パーセント) コマンドを識別し、出力ファイル内にそのまま配置される未加工の PostScript コードの開始を知らせます。**%** (パーセント記号) と **.** (ピリオド) を含む行との間にあるものはすべて (ただし、当該記号は含まない)、作成された PostScript 出力に置かれます。

この PostScript 出力は、**troff** コマンド座標系や作成された PostScript 出力の状況からは分離されません。ただし、希望するならユーザーが分離できるように、2 つの機能がプロローグで定義されます。**PB** (ピクチャー開始) 機能は PostScript 保存操作を行い、PostScript 座標系を、ページ上の現在位置に関する **troff** の概念に変換し、座標系の軸のスケールと向きを標準の、1 インチ当たり PostScript 72 単位へ変更します。**PE** (ピクチャー終了) マクロは、この保護された環境を終了させます。

このように組み込まれた PostScript コードを **troff** 中間ファイルに取り込むには、いくつかの方法を使用できます。例えば、**troff** コマンドの **.sy**、**¥!**、および **.cf** のサブコマンドでは、次の例を使用して、完全に区切られていて印刷可能な文書の PostScript 言語記述を組み込みます。この例では、次に示すように、**showpage** 演算子を再定義して、**mypic.ps** を組み込んでいます。

```
standard troff input
\&
.fl
\!%PB
\!/showpage{}def
.fl
.sy cat mypic.ps
\!PE
\!.
more standard troff input
```

**psdit** コマンドと **enscript** コマンドに対する各種メディア・サイズを含む情報は、ファイル **/usr/lib/ps/MediaSizes** に入っています。

**MediaSizes** ファイルの各エントリーに必要とされる情報は、TranScript で使用した PostScript プリンターに合う **PostScript Printer Description**、つまり **PPD** ファイルから獲得できます。**PPD** ファイルは、Adobe® Systems Incorporated から入手できます。**PPD** ファイルから抽出される測定単位は、ポイントです。プリンターの 1 ポイントは、1 インチの 1/72 です。

ASCII \* (アスタリスク) で始まる **MediaSizes** ファイルのどの行も無視されるのは、**enscript** コマンドと **psdit** コマンドへのコマンド行に提供されたメディア・サイズ名に合致する場合です。

**MediaSizes** ファイルの各エントリーには、8 つか 9 つのフィールドが含まれます。すべてのエントリーについて、最初の 8 フィールドは必要です。9 番目のフィールドはオプションです。フィールドはホワイト・スペースで区切られます。各エントリーのフィールドは、次のとおりです。

フィールド名	説明
<b>EntryName</b>	<b>enscript</b> コマンドか <b>psdit</b> コマンドの <b>-M</b> オプションで提供されるメディア名に一致する文字列。
<b>MediaWidth</b>	ポイント単位のメディアの幅。
<b>MediaDepth</b>	ポイント単位のメディアの深さ。
<b>ImageableLLX</b>	ポイント単位のイメージ可能な左下隅の x 座標。
<b>ImageableLLY</b>	ポイント単位のイメージ可能な左下隅の y 座標。
<b>ImageableURX</b>	ポイント単位のイメージ可能な右下隅の x 座標。
<b>ImageableURY</b>	ポイント単位のイメージ可能な右下隅の y 座標。

フィールド名	説明
<b>PageRegionName</b>	作画可能領域のサイズを識別する特定プリンターのための PostScript シーケンス。
<b>PaperTrayName</b>	特定の紙/メディア・トレイを選択する特定プリンターのための PostScript シーケンス。 このフィールドはオプションです。

注: シーケンスは、**PageRegionName** フィールドと **PaperTrayName** の両フィールドについて、複数の PostScript 演算子やワードにすることができます。このようなシーケンスを指定するには、ASCII " (二重引用符) を使用してシーケンス全体を区切ります。

以下に示すのは、**MediaSizes** ファイルのフィールド・エントリーの例です。

名前	エントリー
<b>Letter</b>	幅 612 縦の長さ 792 <b>llx</b> 18 <b>lly</b> 17 <b>urx</b> 597 <b>ury</b> 776 ページ-領域-名 Letter ページ-トレイ-名
<b>Legal</b>	幅 612 縦の長さ 1008 <b>llx</b> 18 <b>lly</b> 17 <b>urx</b> 597 <b>ury</b> 992 ページ-領域-名 Legal ページ-トレイ-名

## フラグ

**-f1** *CodeSet:Font*

**-F***FontDirectory*

**-M***MediaName*

デフォルトの代わりに *FontDirectory* からフォント情報を取ります。

紙のイメージ可能区域の量を決定するのに使用されるメディア名を指定します。提供される名前は、**MediaSizes** ファイルのエントリーと一致します。例えば、**-M legal** はイメージ可能区域としてリーガル・サイズの紙を要求します。このオプションを使用しない場合、デフォルトのサイズはレターサイズで、幅 8.5 インチで深さ 11.0 インチになります。

**-p**Prologue  
**-o**List

デフォルトの PostScript プロローグの代わりに *Prologue* の内容を使用します。コンマで区切ったリストで提供される番号のページを印刷します。そのリストは、フォーマット *N1-N2* で単一番号と範囲を含み、ここで *N1* と *N2* はページ番号を表します。*N1* が指定されないと、範囲は最小ページから始まり、*N2* が指定されないと、範囲が最大ページで終わることを意味します。

## 例

次の例は等価です。

```
pic -Tpsc File | troff -Tpsc | psc
```

```
pic -Tpsc File | troff -Tpsc | psdit
```

## 環境変数

**PSLIBDIR** **psc** と **psdit** コマンド・プロローグ用 **/usr/lib/ps** ファイルの代わりに使用されるディレクトリの一のパス名。  
**TRANSCRIPT** MBCS ハンドル用 **/usr/lib/ps/transcript.conf** の代わりに使用されるファイルの絶対パス名。

## ファイル

**/usr/lib/font/devpsc/\***  
**/usr/lib/ps/psdit.pro**  
**/usr/lib/ps/MediaSizes**  
**/usr/lib/ps/transcript.conf**

PostScript 仮想デバイスの **troff** デフォルト記述ファイルが入っています。デフォルトの PostScript プロローグが入っています。メディア・サイズに使用されるデフォルト・ファイルが入っています。PostScript のコード・セットとフォント名に使用するデフォルト値が入っています。

## 関連情報

**enscript** コマンド、**managefonts** コマンド、**qprt** コマンド、**pic** コマンド、**psroff** コマンド、**troff** コマンド。

**PE** マクロ。

**nroff** コマンドと **troff** コマンド用 **me** マクロ・パッケージ。

---

## pshare コマンド

### 目的

共用ログイン・ポートを使用可能にするか、使用可能性を報告します。

### 構文

```
pshare [ -a ] [ Device ]
```

### 説明

**pshare** コマンドは、共用ポートを使用可能にします。共用ポートは、両方向です。*Device* パラメーターを指定しない場合には、**pshare** コマンドは、現行で使用可能になっているすべての共用ポートの名前を報告します。共用ポートを使用可能にするために、**getty** コマンドはプロセスの ASCII プロセス ID が入っ



ている **/etc/locks** ディレクトリーの **lock** ファイルを作成しようとします。別のプロセスが既にポートを使用している場合には、ポートが使用可能になるまで **getty** コマンドを待って再度やり直します。システムでは、**/etc/inittab** ファイル内のエントリーを更新し、次に **init** プロセスにシグナルを送信することによって、ポートを使用可能にします。シグナルを受信して更新された状況エントリーを読み込むと、プロセスは、適切なアクションを行います。

*Device* パラメーターを使用して、使用可能にするポートを指定します。以下の値が使えます。

- **F/dev/tty1** デバイスなどの完全デバイス名。
- **tty1** デバイスなど、単純デバイス名。
- 番号 (例えば、**/dev/tty1** デバイスを示す 1 など)。

注: このコマンドを実行するには **root** ユーザー権限を持っていないければなりません。

## フラグ

**-a** すべてのポートを共用として使用可能にします。

## 例

**/dev/tty2** ポートに接続されたワークステーションを共用ポートとして使用可能にするには、次のように入力します。

```
pshare /dev/tty2
```

**/etc/inittab** システムの初期化を制御します。

## ファイル

**/etc/locks** **pshare** および **pdelay** コマンドの **lock** ファイルが入っています。  
**/usr/sbin/pshare** **pshare** コマンドが入っています。

## 関連情報

**getty** コマンド、**init** コマンド、**pdelay** コマンド、**pdisable** コマンド、**penable** コマンド、**phold** コマンド、**pstart** コマンド。

---

## psplot コマンド

### 目的

プロット・フォーマットのファイルを PostScript フォーマットに変換します。

### 構文

```
psplot [ -g Prologue ] [ File... ]
```

### 説明

**psplot** コマンドは、プロット・フォーマットのファイルを読み取り、それを標準出力での PostScript フォーマットに変換します。ファイルを指定しないと、標準入力を使用されます。この変換は、ほぼ 1 対 1 で

行われ、各プロット図形エレメントについて 1 つの PostScript 機能コールによります。プロローグで PostScript 機能の定義を変更することでそのファイルの振る舞いを変更できます。

## フラグ

**-gPrologue** デフォルトの PostScript プロローグの代わりに *Prologue* ファイルの内容を使用します。このフラグを指定しない場合には、デフォルトのプロローグ・ファイルが使用されます。

## 各国文字サポート

文字変換についての詳細は、**NLSvec** ファイルを参照してください。

## 環境変数

**PSLIBDIR** **psplot** コマンドのプロローグ・ファイルに使用されるディレクトリーのパス名。 **PSLIBDIR** 環境変数によって指示されるパスが、**XPSLIBDIRX** 環境変数によって指示されるパスの代わりに使用されます。

**TRANSCRIPT** MBCS ハンドル用 **/usr/lib/ps/transcript.conf** の代わりに使用されるファイルの絶対パス名。

## ファイル

**/usr/lib/ps/psplot.pro** デフォルトのプロローグ・ファイルが入っています。

## 関連情報

**ps4014** コマンド、 **lpr** コマンド、 **lp** コマンド。

**NLSvec** ファイル。

---

## psrev コマンド

### 目的

PostScript ファイルのページ順序を逆にし、印刷のページ範囲を選択します。

### 構文

```
psrev [ -R ] [ -sPagespec,... ] [ File ]
```

### 説明

**psrev** コマンドは、*File* 変数で指定したファイルのページ順序を逆にし、*Pagespec* パラメーターで指定したページを印刷します。このファイルは、PostScript ファイル構造基準に合致していなければなりません。*File* に値を指定しないと、**psrev** コマンドは標準入力から読み取ります。**psrev** コマンドは、結果として生成されたファイルを標準出力へ書き出します。

## フラグ

**-R** ページ順序を逆にしません (指定すれば、ページをサブセットします)。

**-sPagespec** 印刷すべきページの範囲 (または複数の範囲) を指定します。 *Pagespec* パラメーターはスペースなしの文字列です。 *Pagespec* パラメーターは、単一のページ番号または *N-M* フォーマットの範囲にすることができ、*N* から *M* までのページを印刷します。 *-N* は、文書の最初からページ *N* までを印刷します。 *-M* は、ページ *M* から文書の終わりまでを印刷します。

## 例

次は、ページ範囲と逆順しない個別ページを示した **psrev** コマンドの使用例です。

```
psrev -R -s2-4,6
```

```
psrev -R -s2-4,6-8
```

## ファイル

**/var/tmp/RV\*** 入力がパイプ接続である場合の一時ファイルが入っています。

## 関連情報

**enscript** コマンド。

---

## psroff コマンド

### 目的

**troff** フォーマットのファイルを PostScript フォーマットに変換します。

### 構文

```
psroff [ -t ] [ -dQueue ] [ -nNumber ] [ -tTitle ] [ -DFontDirectory ] [ -FFontFamily ] [ -PFlag ]  
[ troffFlags ] [ File ... ]
```

### 説明

**psroff** コマンドは、PostScript プリンターに出力を生成する環境で **troff** コマンドを実行するシェル・スクリプトです。このコマンドは、**psdit** コマンドを使用して **troff** 中間出力を PostScript フォーマットに変換し、印刷できるようにこれをスプールします。ファイルを指定しないと、標準入力を使用されます。

任意 PostScript 言語コマンドやファイルを **troff** 文書に組み込むには、**psdit** コマンドのセクションを参照してください。

### PostScript フォント情報

Transcript テーブルの PostScript フォントは、TranScript コマンドに使用できるフォントを示します。このフォントは、**enscript** コマンドを使用するときは長名で使用でき、**psroff** コマンドか **troff** コマンドを使用するときは省略名で使用できます。次のテーブルは、フォントのデフォルト・セットを宣言するとき使用する **psroff** コマンド (省略名) です。英文字は大文字小文字を区別します。

トランスクリプト用の PostScript フォント

ロング・ネーム (ショート・ネーム)	フォント・ファミリー
AvantGarde-Book (ag)	AvantGarde
AvantGarde-Demi (Ag)	AvantGarde

トランスクリプト用の PostScript フォント

ロング・ネーム (ショート・ネーム)	フォント・ファミリー
AvantGarde-DemiOblique (AG)	AvantGarde
AvantGarde-BookOblique (aG)	AvantGarde
Bookman-Demi (Bo)	Bookman
Bookman-DemiItalic (BO)	Bookman
Bookman-Light (bo)	Bookman
Bookman-LightItalic (bO)	Bookman
Courier (C)	Courier
Courier-Bold (CB)	Courier
Courier-BoldOblique (CO)	Courier
Courier-Oblique (CO)	Courier
Garamond-Bold (Ga)	Garamond
Garamond-BoldItalic (GA)	Garamond
Garamond-Light (ga)	Garamond
Garamond-LightItalic (gA)	Garamond
Helvetica (H)	Helvetica
Helvetica-Bold (HB)	Helvetica
Helvetica-Oblique (HO)	Helvetica
Helvetica-BoldOblique (HD)	Helvetica
Helvetica-Narrow (hn)	Helvetica
Helvetica-Narrow-Bold (Hn)	Helvetica
Helvetica-Narrow-BoldOblique (HN)	Helvetica
Helvetica-Narrow-Oblique (hN)	Helvetica
LubalinGraph-Book (lu)	Lubalin
LubalinGraph-BookOblique (IU)	Lubalin
LubalinGraph-Demi (Lu)	Lubalin
LubalinGraph-DemiOblique (LU)	Lubalin

NewCenturySchlbk (NC)	NewCentury
NewCenturySchlbk-Bold (Nc)	NewCentury
NewCenturySchlbk-Italic (nC)	NewCentury
NewCenturySchlbk-Roman (nc)	NewCentury
Optima (op)	Optima
Optima-Bold (Op)	Optima
Optima-BoldOblique (OP)	Optima
Optima-Oblique (oP)	Optima
Palatino-Bold (PB)	Palatino
Palatino-BoldItalic (PX)	Palatino
Palatino-Italic (PI)	Palatino
Palatino-Roman (PA)	Palatino
Souvenir-Demi (Sv)	Souvenir

Souvenir-DemiItalic (SV)	Souvenir
Souvenir-Light (sv)	Souvenir
Souvenir-LightItalic (sV)	Souvenir
Times-Bold (TB)	Times
Times-BoldItalic (TD)	Times
Times-Italic (TI)	Times
Times-Roman (TR)	Times
Symbol (S)	(なし)
ZapfChancery-MediumItalic (ZC)	Zapf
ZapfDingbats	(なし)

## フラグ

### **-D***FontDirectory*

標準フォント・ディレクトリーではなく、インストール手順で構成された指定フォント・ディレクトリーでフォント・ファミリー・ディレクトリーを検索します。 **troff** コマンドの **-F** フラグをまねるには、このフラグと **-F** フラグの両者を使用することが必要な場合があります。

### **-d***Queue*

*Queue* パラメーターで指定したキューに出力を入れます。 **-d** フラグを使用しない場合には、**psroff** コマンドが出力をデフォルトのキューに入れ、最初のキューが **qdaemon** に対して認識されます。このフラグは、スプーラー印刷によって認識されます。

### **-F***FontFamily*

Times デフォルト・ファミリーではなく、R/I/B/BI フォント用の指定フォント・ファミリーを使用します。Times、Courier、Helvetica フォント・ファミリーはユーザーの現場で定義され、他のフォントも同様に使用できます。ユーザーが使用するプリンターが、ユーザーの選ぶフォント・ファミリーを持っていることを確認してください。このフラグは、**troff** コマンドの **-F** フラグを変更します。 **troff** コマンドの **-F** フラグを使用したい場合には、直接 **troff** コマンドを実行するか、代わりに **-D** フラグを使用する必要があります。

### **-n***Number*

*Number* パラメーターで指定された出力のコピー数を作成させます。デフォルトは 1 部です。このフラグは、スプーラー印刷によって認識されます。

### **-P***Flag*

*Flag* パラメーターをスプーラーへ渡します。これは、スプーラーのフラグと **psroff** コマンドを使用したフラグの間で矛盾がある場合に便利です。

### **-t**

PostScript 出力をプリンターへスプールするのではなく標準出力を送信します。このフラグは、**troff** コマンドの **-t** フラグを変更します。 **troff** コマンドの **-t** フラグを使用したい場合には、直接 **troff** コマンドを実行する必要があります。

### **-t***Title*

最初のパナー・ページで使用するジョブ名を設定します。デフォルトでは、最初の入力ファイルの名前が使用されます。このフラグは、スプーラー印刷によって認識されません。

## パラメーター

### *troffFlags*

**troff** コマンドで使用可能な標準フラグを指定します。

### *File*

**troff** 中間出力ファイルを指定します。デフォルトは標準入力です。

## ファイル

### **/usr/share/lib/tmac/tmac.\***

標準マクロ・ファイルが入っています。

### **/usr/lib/font/devpsc/\***

PostScript 仮想デバイスの **troff** 記述ファイルが入っています。

`/usr/lib/ps/*.afm`

**enscript** コマンドで使用するための Adobe フォント・メトリック (AFM) ファイルが入っています。

`/usr/lib/ps/font.map`

フォント名と省略形のリストが入っています。

`/usr/lib/ps/ditroff.font`

**troff** コマンドのフォント・ファミリー・ファイルが入っています。

## 関連情報

**col** コマンド、**enscript** コマンド、**eqn** コマンド、**lp** コマンド、**managefonts** コマンド、**nroff** コマンド、**pic** コマンド、**psdit** コマンド、**refer** コマンド、**tbl** コマンド、**troff** コマンド。

---

## pstart コマンド

### 目的

ログイン・ポート (通常、共用、および遅延) を使用可能にするか、使用可能性を報告します。

### 構文

**pstart** [ **-a** ] [ *Device* ]

### 説明

**pstart** コマンドは、**/etc/inittab** ファイルにリストされているすべてのポート (通常、共用、および遅延) を使用可能にします。システムでは、**/etc/inittab** ファイル内のエントリを更新し、次に **init** プロセスにシグナルを送信することによって、ポートを使用可能にします。**init** プロセスがシグナルを受信して、更新された状況エントリを読み込んだとき、適切なアクションを行います。

*Device* パラメーターを使用して、使用可能にするポートを指定します。以下の値が使えます。

- **/dev/tty1** デバイスなどの完全デバイス名。
- **tty1** デバイスなどの単純デバイス名。
- 番号 (例えば、**/dev/tty1** デバイスを示す 1 など)。

*Device* パラメーターを指定しない場合には、**pstart** コマンドは、すべての使用可能ポートの名前、および現行でそれらのポートが通常、共用、遅延のどれかで使用可能になっているのかを報告します。

注: このコマンドを実行するには root ユーザー権限を持っていないければなりません。

### フラグ

**-a** すべてのポート (通常、共用、および遅延ポート) を使用可能にします。

### 例

1. 現在使用可能になっているすべてのポート (通常、共用、および遅延) の名前と、その使用可能状態を表示するには、次のように入力します。

```
pstart
```

2. **/etc/inittab** ファイル内にリストされた通常、共用、および遅延ポートをすべて使用可能にするには、次のように入力します。

```
pstart -a
```

## ファイル

`/etc/locks`  
`/usr/sbin/pstart`

`pshare` および `pdelay` コマンドの `lock` ファイルが入っています。  
`pstart` コマンド・ファイルが入っています。

## 関連情報

`init` コマンド、`pdelay` コマンド、`pdisable` コマンド、`penable` コマンド、`phold` コマンド、`pshare` コマンド。

`inittab` ファイル。

---

## pstat コマンド

### 目的

各種システム・テーブルの内容を解釈して、標準出力に書き出します。

### 構文

```
pstat [ -a ][ -A ][ -f ][ -i ][ -p ][ -P ][ -s ][ -S ][ -t ][ -uProcSlot ][ -T ]  
[ -U ThreadSlot ] [ [ KernelFile ] CoreFile ]
```

### 説明

`pstat` は、各種システムのテーブルの内容を解釈して、標準出力を書き込みます。`pstat` コマンドを実行するためには、`root` ユーザーまたは `system` グループの権限を持っていないければなりません。

### フラグ

<code>-a</code>	プロセス・テーブル内のエントリを表示します。
<code>-A</code>	カーネル・スレッド・テーブル内のエントリをすべて表示します。
<code>-f</code>	ファイル・テーブルを表示します。
<code>-i</code>	<code>i</code> ノード・テーブルと <code>i</code> ノード・データ・ブロックのアドレスを表示します。
<code>-p</code>	プロセス・テーブルを表示します。
<code>-P</code>	実行可能なカーネル・スレッド・テーブル・エントリのみを表示します。
<code>-s</code>	スワップまたはページング・スペースの使用に関する情報を表示します。
<code>-S</code>	プロセッサの状況を表示します。
<code>-t</code>	<code>tty</code> 構造体を表示します。
<code>-u ProcSlot</code>	プロセス・テーブル内の指定されたスロット内のプロセスのユーザー構造体を表示します。スワップアウトされたプロセスを表示しようとするエラー・メッセージが生成されます。
<code>-T</code>	システム変数を表示します。これらの変数は、 <code>var.h</code> に簡単に説明されています。
<code>-U ThreadSlot</code>	カーネル・スレッド・テーブル内の指定されたスロット内のカーネル・スレッドのユーザー構造体を表示します。スワップアウトされたカーネル・スレッドを表示しようとするエラー・メッセージが生成されます。

### 例

1. `dumpfile` コア・ファイルに保存されているシステム・ダンプの `i` ノードを表示するには、次のように入力します。

```
pstat -i dumpfile
```



シンボルは、**/usr/lib/boot/unix** ファイルから検索されます。

2. システムが現在実行中のプロセス・テーブル・スロット 0 (ゼロ) のプロセスについて、ファイル・テーブルとユーザー構造体を表示するには、次のように入力します。

```
pstat -f -u 0
```

3. コア・ファイルが **dumpfile** であって、カーネルが **/usr/lib/boot/unix.back** ファイルであるようなシステム・ダンプの tty 構造体を表示するには、次のように入力します。

```
pstat -t /usr/lib/boot/unix.back dumpfile
```

4. カーネル・スレッド・テーブル内のすべてのスレッドおよびスレッド・テーブル・スロット 2 内のスレッドのユーザー構造体を表示するには、次のように入力します。

```
pstat -A -U 2
```

## ファイル

**/usr/sbin/pstat**  
**/dev/mem**  
**/usr/lib/boot/unix**  
**/usr/include/sys/\*.h**

**pstat** コマンドが入っています。  
デフォルトのシステム・イメージ・ファイル。  
デフォルトのカーネル・イメージ・ファイル。  
テーブルおよび構造情報用のヘッダー・ファイルが入っています。

## 関連情報

**ps** コマンド、**stty** コマンド。

---

## ptx コマンド

### 目的

順列索引を生成します。

### 構文

```
ptx [ -f ] [ -r ] [ -t ] [ -b Breakfile ] [ -g Number ] [ -w Number ] [ -i Ignore | -o Only ] [ - ]  
[ Infile [ Outfile ] ]
```

### 説明

**ptx** コマンドは、指定された英語テキスト (*Infile* パラメーター) を読み取り、そこから配列し直した索引を作成し、指定されたファイル (*Outfile*) へ書き込みます。標準入力と標準出力がデフォルトです。

**ptx** コマンドは、指定されたファイル (*Infile*) でキーワードを探索し、行をソートし、ファイル *Outfile* を作成します。そこで、*Outfile* ファイルを **nroff** または **troff** のコマンドで処理し、再配列した索引を作成できます。

**ptx** コマンドは以下の 3 つのステップで処理します。

1. 順列化を実行し、入力行内の各キーワードごとに 1 行ずつ生成して、キーワードを行の前に回転させます。
2. 順列ファイルをソートします。
3. キーワードが各行の中央になるようにソートされた行を回転させます。

結果として生成される行のフォーマットは、*Outfile* ファイル内で次のようになります。

```
.xx "" "before keyword" "keyword" "after keyword"
```

この場合、.xx は、ユーザーまたは **ptx** コマンドが与える **nroff** または **troff** マクロです。 **mptx** マクロ・パッケージは、.xx マクロ定義を提供します。

before keyword、keyword、および after keyword フィールドは、出力時にキーワードの前後に入る行数として組み込まれます。先頭フィールドと最終フィールドは、少なくとも一方は常に空文字列で、行の反対端の未使用のスペースに入るように折り返されます。

#### 注:

1. 行の長さの数は、重ね打ちスペーシングまたはプロポーショナル・スペーシングを計算に入れません。
2. ~ (ティルド) を含む行が機能しないのは、**ptx** コマンドがその文字を内部で使用するためです。
3. **ptx** コマンドは英数字以外の文字を破棄しません。

## フラグ

<b>-b</b> <i>BreakFile</i>	指定された切断ファイル内の文字を使用してワードを区切ります。タブ文字、改行文字、スペースは、常に切断文字として使用されます。
<b>-f</b>	ソートのときに大文字と小文字を区別しません。
<b>-g</b> <i>Number</i>	出力されるときに <b>ptx</b> コマンドが行の 4 つの部分間でそれぞれのギャップに予約する文字として、指定数を使用します。デフォルトの <i>Number</i> 変数値は 3 です。
<b>-i</b> <i>Ignore</i>	<i>Ignore</i> ファイルに指定されたワードはキーワードとして使用しません。 <b>-i</b> と <b>-o</b> フラグを使用しない場合には、 <b>/usr/lib/eign</b> ファイルがデフォルトの <i>Ignore</i> ファイルです。
<b>-o</b> <i>Only</i>	<i>Only</i> ファイルに指定されたワードだけをキーワードとして使用します。
<b>-r</b>	各入力行のブランク以外の先頭文字を、行のテキストと切り離された参照 ID と見なします。ID を、各出力行の 5 番目のフィールドとして接続します。
<b>-t</b>	フォントタイプセッター用の出力を準備します。
<b>-w</b> <i>Number</i>	出力行の長さとして、指定された数を使用します。デフォルトの行長は、 <b>nroff</b> コマンドの場合には 72 文字で、 <b>troff</b> コマンドの場合には 100 文字です。
—	(二重ダッシュ) フラグの終わりを示します。

## パラメーター

<i>Infile</i>	英語テキストを指定します。標準入力がデフォルトのファイルです。 <b>ptx</b> コマンドは、指定ファイルでキーワードを検索し、行をソートし、ファイル <i>Outfile</i> を作成します。
<i>Outfile</i>	<b>ptx</b> コマンドが <i>Infile</i> ファイルから作成した索引を書き込むファイルを指定します。標準出力がデフォルトのファイルです。 <i>Outfile</i> ファイルを <b>nroff</b> または <b>troff</b> のコマンドで処理し、再配列した索引を作成できます。

## ファイル

<b>/usr/lib/eign</b>	デフォルトの <i>Ignore</i> ファイルが入っています。
<b>/usr/share/lib/tmac/tmac.ptx</b>	マクロ・ファイルが入っています。

## 関連情報

**nroff** コマンド、 **troff** コマンド。

**mm** マクロ・パッケージ、 **mptx** マクロ・パッケージ。

---

## pwchange コマンド

### 目的

ユーザー認証およびプライバシー・キーを動的に変更します。

### 構文

```
pwchange [ -e ] [ -d DebugLevel ] [ -p Protocol ] [ -u KeyUsage ] [ -s ] [ OldPassword NewPassword ]  
[ IPAddress | HostName | EngineID ]
```

### 説明

**pwchange** コマンドは、ユーザー認証およびプライバシー・キーを動的に変更できるようにするために提供されています。認証およびプライバシー・キーの動的な構成は、構文 **KeyChange** のオブジェクトに **set** コマンドを実行して行います。実際のキー (新しいキーまたは古いキー) を回線に直接転送するのは安全でないため、**KeyChange** 構文は、そのような転送を必要としないでキーを変更する方法を提供します。実際のキーを回線に直接転送するのではなく、例えばオブジェクト **usmUserAuthKeyChange** (例) が設定されていた場合には、**keyChange** 値は、古いパスワードと新しいパスワードから導き出す必要があり、キーが使用されるエージェントの **engineID** が使用されます。**pwchange** コマンドは、**keyChange** 値の生成に使用されます。

**pwchange** コマンドは、選択したプロトコルとキー用途により、異なる出力を生成します。通常 **Keychange** 値は、変更されるキーの 2 倍の長さです。

### フラグ

**-d** *DebugLevel*

このフラグは、デバッグ情報の望ましいレベルを指定します。デバッグ・トレースは、オンまたはオフのいずれかであり、1 は、デバッグ・トレースをコマンド実行者の画面 (sysout) に生成します。デフォルトでは、デバッグ・トレースはオフ (0) です。

**-e**

このフラグは、**keychange** 値を定義するエージェントを、IP アドレスまたはホスト名ではなく、**engineID** で識別することを示します。

**-p** *Protocol*

このフラグは、**keychange** 値を生成すべきプロトコルを指定します。有効な値は以下のとおりです。

#### HMACH-MD5

HMACH-MD5 認証プロトコルで使用する **keychange** 値を生成する。

#### HMACH-SHA

HMACH-SHA 認証プロトコルで使用する **keychange** 値を生成する。

**all** HMACH-MD5 および HMACH-SHA の両方の **keychange** 値を生成する。

デフォルトでは、HMACH-MD5 プロトコルの **keychange** 値が生成されます。

**-s**

このフラグは、読みやすくするために、スペースを追加して出力を表示することを指定します。デフォルトでは、データはスペースのないフォーマットで表示されますが、その理由は、keychange 値をシェル・スクリプトのコマンド行にカット・アンド・ペースト操作できるようにするためです。

**-u** *KeyUsage*

このフラグは、keychange 値に意図した用途を示します。有効な値は以下のとおりです。

**auth** 認証 keychange 値。

**priv** プライバシー keychange 値。

**all** 認証およびプライバシーの両方の keychange 値。  
**注:** 認証用に生成される keychange 値とプライバシー用に生成される keychange 値には、違いはありません。ただし、プライバシー keychange 値の長さは、keychange 値をローカライズするかどうかにより異なります。

## パラメーター

*EngineID*

キーを使用する宛先ホストの engineID (1 から 32 のオクテット、2 から 64 の 16 進数) を指定します。

*HostName*

engineID は 1 から 32 のオクテット (2 から 64 の 16 進数) の文字列でなければなりません。デフォルトでは、エージェント ID は engineID ではありません。

*IPAddress*

鍵を使用する宛先ホストを指定します。

*NewPassword*

キーを使用する宛先ホスト側のエージェントの、ドット 10 進数表記の IP アドレスを指定します。

*OldPassword*

新しい鍵の生成に使用されるパスワードを指定します。パスワードの長さは 8 から 255 文字の間にする必要があります。

元の鍵の生成に使用されたパスワードを指定します。パスワードの長さは 8 から 255 文字の間にする必要があります。

## 例

**pwchange** コマンドは、選択したプロトコルとキー用途により、異なる出力を生成します。通常 Keychange 値は、変更されるキーの 2 倍の長さです。

1. 以下のコマンドは pwchange の使用法を示しています。

```
pwchange oldpassword newpassword 9.67.113.79
```

このコマンドの出力は以下のようなものです。

```
Dump of 32 byte HMAC-MD5 authKey keyChange value:  
3eca6ff34b59010d262845210a401656  
78dd9646e31e9f890480a233dbe1114d
```

設定する値は、**clsnmp** コマンドを使用して、16 進値として渡す必要があります (すべてを 1 行に指定)。

```
clsnmp set usmUserAuthKeyChange.12.0.0.0.2.0.0.0.9.67.113.79.2.117.49
\'3eca6ff34b59010d262845210a40165678dd9646e31e9f890480a233dbe1114d\'h
```

注: AIX が 16 進値を正しく解釈できるようにするため、上の例では単一引用符の前にバックスラッシュ (\) が必要です。

usmUserTable の索引は、EngineID とユーザー名の ASCII 表現からなります。この例の場合は、長さが 2 文字で 117.49 に変換されます。

注: **pwchange** は、キーおよび keyChange 値の生成にランダム・コンポーネントを採用しています。したがって、同じ入力を使用する複数のコマンドは同じ結果にはなりません。

## 関連情報

**clsnmp** コマンド、**pwtokey** コマンド、**snmpdv3** デーモン。

**/etc/clsnmp.conf** ファイル、**/etc/snmpdv3.conf** ファイル。

---

## pwck コマンド

### 目的

ローカル認証情報が正しいかどうかを検証します。

### 構文

**pwck**

### 説明

**pwck** コマンドは、すべてのユーザーの定義を検査し、ユーザー・データベース・ファイル内のパスワード情報が正しいことを検証します。**pwck** コマンドは内部的に、**-n** および **ALL** オプションを使用して **pwdck** コマンドを呼び出します。

### 終了状況

0 コマンドは正常に実行されました。

>0 エラーが発生しました。

### 例

- すべてのユーザーおよび管理者がユーザー・データベースに存在することを検証し、エラーを報告させる (しかし、修正はしない) には、次のように入力します。

```
pwck
```

### ファイル

**/usr/bin/pwck**

**pwck** コマンドが入っています。

### 関連情報

**pwdck** コマンド。

---

## pwd コマンド

### 目的

作業ディレクトリーのパス名を表示します。

### 構文

```
pwd [ -L | -P ]
```

### 説明

**pwd** コマンドは、現行ディレクトリーの絶対パス名 (ルート・ディレクトリーから) を標準出力に書き出します。すべてのディレクトリーは / (スラッシュ) で区切られます。ルート・ディレクトリーは先頭の / で表され、最後のディレクトリーが現行ディレクトリーです。

### フラグ

- L** ファイル名 . (ドット) または .. (ドット・ドット) を含んでいない現行ディレクトリーの絶対パス名が **PWD** 環境変数に含まれている場合は、**PWD** 環境変数の値を表示します。それ以外の場合には、**-L** フラグは **-P** フラグと同じ働きをします。
- P** 現行ディレクトリーの絶対パス名を表示します。 **-P** フラグを使用して表示される絶対パス名には、シンボリック・リンクのタイプのファイルを参照する (パス名のコンテキストの) ファイル名が含まれません。

### 終了状況

このコマンドは、以下の終了値を戻します。

- 0** 正常終了。
- >0** エラーが発生しました。

### 例

入力

```
pwd
```

現行ディレクトリーが次のように表示されます。

```
/home/thomas
```

### ファイル

```
/usr/bin/pwd
```

**pwd** コマンドが入っています。

### 関連情報

**cd** コマンド。

**getcwd** サブルーチン。

オペレーティング・システムおよびデバイスのマネージのディレクトリーのセクション。

---

## pwdadm コマンド

### 目的

ユーザーのパスワードを管理します。

### 構文

```
pwdadm [ -R load_module ] [ -f Flags | -q | -c ] User
```

### 説明

**pwdadm** コマンドは、ユーザーのパスワードを管理します。 **root** ユーザーやセキュリティ・グループのメンバーが、**User** パラメーターで指定されたユーザーのパスワードを提供したり変更したりできます。コマンドを起動したユーザーは、他のユーザーのパスワードの変更が許可される前に、パスワードを照会されたら提供しなければなりません。コマンドは実行されるときに **ADMCHG** 属性を設定します。そのため、**login** コマンドや **su** コマンドがユーザーに与えられる次のときに、ユーザーはパスワードを変更する必要があります。

**注:** このコマンドに関して説明する動作は、ローカル・ユーザーの場合を対象にしています。リモート・ドメインに定義されているユーザーの場合は、属性はローカル・ファイルではなくリモート・ドメインで検索され保管されることとなります。

**root** ユーザーまたはセキュリティ・グループのメンバーは、このコマンドを使用して自分の個人用パスワードを変更してはなりません。その理由は、次回そのユーザーに対して **login** コマンドまたは **su** コマンドが指定されたときに、**ADMCHG** 属性があるためにパスワードを再度変更する必要があるからです。**root** ユーザー、または、**PasswdAdmin** 権限が与えられているユーザーだけが、**/etc/security/user** ファイル内で **admin** 属性に **true** を設定されている管理ユーザーに関するパスワード情報を変更できます。

**root** ユーザー、セキュリティ・グループのメンバー、または、**PasswdManage** 権限が与えられているユーザーだけが、**User** パラメーターで指定されたユーザーのパスワードを提供したり変更したりすることができます。

このコマンドを実行するときには、**/etc/passwd** ファイルのユーザー用 **password** フィールドが ! (感嘆符) に設定され、パスワードの暗号化バージョンが **/etc/security/passwd** ファイルにあることを示します。**ADMCHG** 属性が設定されるのは、**root** ユーザーやセキュリティ・グループのメンバーが **pwdadm** コマンドでユーザーのパスワードを変更するときです。

**-f NOCHECK** フラグを指定しない限り、新しいパスワードは、**/etc/security/user** ファイル内の規則にしたがって定義しなければなりません。7 ビット文字だけがパスワードでサポートされます。**pwdadm** コマンドに **-f** フラグを指定すると、**root** ユーザーまたはセキュリティ・グループのメンバーはパスワード規則を変更する属性を設定できます。**-f** フラグを使用するとき **/etc/security/passwd** ファイル内にパスワード・エントリがなければ、**/etc/passwd** ファイルの **password** フィールドが ! (感嘆符) に設定され、パスワードが設定されていないことを示す \* (アスタリスク) が **password=** フィールドに表示されます。

**-q** フラグを使用すると、**root** ユーザーまたはセキュリティ・グループのメンバーはパスワード情報を照会できます。**lastupdate** 属性と **flags** 属性の状況のみが表示されます。暗号化されたパスワードは表示されません。

**-c** フラグは、ユーザーのすべてのパスワード・フラグをクリアします。



## フラグ

- c** ユーザーのすべてのパスワード・フラグをクリアします。
- f *Flags*** パスワードの **flags** 属性を指定します。 *Flags* 変数は、以下のリストからの属性をコマンドで区切ったリストでなければなりません。
- NOCHECK**  
新しいパスワードがパスワード構成について **/etc/security/user** ファイル内で設定されたガイドラインに従わなくてもよいことを示します。
- ADMIN** root ユーザーしかパスワード情報を変更できないことを示します。この属性を使用可能または使用不可にできるのは root ユーザーのみです。
- ADMCHG**  
ユーザーのパスワードを変更せずに **ADMCHG** 属性をリセットします。これにより、ユーザーは次に **login** コマンドまたは **su** コマンドが与えられるとパスワードを変更することになります。 *User* パラメーターで指定されたユーザーがパスワードをリセットすると、この属性はクリアされます。
- q** パスワードの状況を照会します。 **lastupdate** 属性と **flags** 属性の値が表示されます。
- R *load\_module*** ユーザーの属性を変更するために使用されるロード可能 I&A モジュールを指定します。

## セキュリティ

アクセス制御: root ユーザーとセキュリティ・グループのメンバーのみが、このコマンドへの実行 (x) アクセス権を持ちます。このコマンドには **トラステッド・コンピューティング・ベース** 属性があり、root ユーザーが **/etc/passwd** ファイル、**/etc/security/passwd** ファイル、および他のユーザーのデータベース・ファイルへの書き込みアクセス (w) を持つように **setuid** します。

アクセスされるファイルは次のとおりです。

モード	ファイル
rw	<b>/etc/passwd</b>
rw	<b>/etc/security/passwd</b>
r	<b>/etc/security/user</b>

監査イベントは次のとおりです。

イベント	情報
<b>PASSWORD_Change</b>	ユーザー
<b>PASSWORD_Flags</b>	ユーザー、フラグ

## 例

1. ユーザー **susan** のパスワードを設定するには、セキュリティ・グループのメンバーが次のように入力します。

```
pwdadm susan
```

プロンプトが出されたら、コマンドを起動したユーザーにパスワードを入力するようにプロンプトが出された後で、**Susan** のパスワードを変更できます。

2. ユーザー **susan** のパスワード状況を照会するには、セキュリティ・グループのメンバーが次のように入力します。

```
pwdadm -q susan
```

このコマンドは、**lastupdate** 属性と **flags** 属性の値を表示します。次の例では、**NOCHECK** と **ADMCHG** の **flag** 属性が有効になったときに表示される内容を示しています。

```
susan:
      lastupdate=
      flags= NOCHECK,ADMCHG
```

## ファイル

<b>/usr/bin/pwddadm</b>	<b>pwddadm</b> コマンドが入っています。
<b>/etc/passwd</b>	基本的なユーザー属性が入っています。
<b>/etc/security/passwd</b>	パスワード情報が入っています。
<b>/etc/security/login.cfg</b>	構成情報が入っています。

## 関連情報

**passwd** コマンド。

ユーザーの識別と認証、任意アクセス制御、トラステッド・コンピューティング・ベース、および監査の詳細については、セキュリティのネットワークの保護のセクションを参照してください。

---

## pwdck コマンド

### 目的

ローカル認証情報が正しいかどうかを検証します。

### 構文

```
pwdck { -p | -n | -t | -y } { ALL | User ... }
```

### 説明

**pwdck** コマンドは、ユーザー **ALL** または *User* パラメーターで指定されたユーザーの定義を検査して、ユーザー・データベース・ファイル内のパスワード情報が正確であるかどうかを検証します。複数のユーザーを指定する場合は、各ユーザー名をスペース 1 つで区切らなければなりません。

注: このコマンドは、**stderr** にメッセージを書き込みます。

システムが間違った属性を修正すべきかどうかを指示するフラグを選択しなければなりません。

**/etc/passwd** ファイル内ではローカル定義ユーザーについて次の属性が検査されます。

<b>entry</b>	各エントリーが読み取り可能で、少なくとも 2 つの : (コロン) を含んでいることを保証します。システムがエラーを修正すべきであると指示した場合には、エントリー全体が廃棄されます。
<b>passwd</b>	パスワード・フィールドが単一の ! (感嘆符) になっていることを保証します。システムがエラーを修正すべきであると指示すると、パスワード・フィールドの情報を <b>/etc/security/passwd</b> ファイルに転送し、 <b>/etc/security/passwd</b> ファイルで <b>lastupdate</b> 属性を変更して、 <b>/etc/passwd</b> ファイルのパスワード・フィールドを単一の ! で置き換えます。一般に、 <b>/etc/security/user</b> ファイルで <b>minalpha</b> 、 <b>minother</b> 、または <b>minlen</b> パスワード制約事項がゼロ以外の値に設定されている場合は、パスワードが必要です。

**user** ユーザー名が 8 バイトかそれ以下の一意の文字列であることを保証します。ユーザー名を + (プラス記号)、: (コロン)、- (マイナス記号)、または ~ (ティルド) で始めることはできません。この文字列には、: (コロン) を含めることができず、**ALL**、**default**、または \* キーワードにすることもできません。システムがエラーを修正すべきであると指示した場合には、このユーザーのエントリー行を **/etc/passwd** ファイルから取り除きます。ユーザー名が + や - 記号で始まる場合には、ユーザーはローカルで定義されず、検査は行われません。

**/etc/security/passwd** ファイル内で検査される属性は、次のとおりです。

**line** 各行が読み取り可能で、スタンザの一部であることを保証します。無効な行があれば破棄されます。

**password** **password** 属性が存在していることと、システム上でパスワードが必要な場合は空白でないことを保証します。システムがエラーを修正すべきであると指示した場合には、パスワードは \* (アスタリスク) に設定され、**lastupdate** 属性が破棄されます。

一般に、**/etc/security/user** ファイルで **minalpha** または **minother** パスワード・リソースがゼロ以外の値に設定されている場合は、パスワードが必要です。ユーザーの **flags** 属性が **NOCHECK** キーワードを指定すると、このユーザーについてはパスワードが不要になり、検査は無視されます。

**lastupdate** 有効な非空白のパスワードについて **lastupdate** 属性が存在することと、その時刻が現在の時刻よりも前であることを保証します。システムがエラーを修正すべきであると指示すると、**password** 属性に応じて **lastupdate** 属性が破棄または更新されます。**password** 属性が存在しないか、空白または \* (アスタリスク) に等しければ、**lastupdate** 属性が破棄されます。そうでなければ、**lastupdate** の時刻が現在の時刻に設定されます。

**flags** **flags** 属性にキーワード **ADMIN**、**ADMCHG**、および **NOCHECK** のみが入っていることを保証します。システムがエラーを修正すべきであると指示すると、未定義のフラグが削除されます。

**/etc/security/user** ファイル内で検査されている属性は、次のとおりです。

**auth1** ローカル・ユーザーについて定義された各 **SYSTEM;username** エントリーが **/etc/security/passwd** ファイル内に **username** エントリーを持っていることを保証します。システムがエラーを修正すべきであると指示すると、存在しないエントリーごとに **/etc/security/passwd** ファイルに以下のフォーマットでスタンザが追加されます。

```
username:  
    password = *
```

ユーザーのエントリーとデフォルトのエントリーの両方が **/etc/security/user** ファイルに存在していないと、システムは以下の値を想定して **auth1** に関する検査を実行します。

```
auth1 = SYSTEM;user
```

**注:** **auth1** 属性は推奨されないもので、使用しないでください。

**auth2** ローカル・ユーザーについて定義された各 `authnameusername` エントリーが `/etc/security/passwd` ファイル内に `username` エントリーを持っていることを保証します。システムがエラーを修正すべきであると指示すると、存在しないエントリーごとに 1 つずつエントリーが追加されます。

ユーザーのエントリーとデフォルトのエントリーの両方が `/etc/security/user` ファイルに存在していないと、システムは以下の値を想定して **auth2** に関する検査を実行します。

```
auth2 = NONE
```

ALL を指定すると、**pwdck** コマンドは `/etc/security/passwd` ファイル内の各スタンザが `/etc/security/user` ファイル内の `SYSTEM;username` エントリーとしてローカル・ユーザーの認証名に対応していることを保証します。システムがエラーを修正すべきであると指示すると、`/etc/security/user` ファイル内の `username` エントリーに対応しないスタンザが、`/etc/security/passwd` ファイルから破棄されません。

**pwdck** コマンドは、更新時に `/etc/passwd` ファイルと `/etc/security/passwd` ファイルをロックします。この 2 つのファイルのどちらかが別のプロセスによってロックされると、**pwdck** コマンドはファイルがアンロックされるのを数分間待機して、この状態が発生しなければ終了します。

**pwdck** コマンドは、現在の **pwdck** プロセスの実行中に `/etc/passwd` ファイルと `/etc/security/passwd` ファイルが別のプロセスによって変更されるかどうかを検査します。システムがエラーを修正すべきであると指示すると、**pwdck** コマンドは `/etc/passwd` ファイルと `/etc/security/passwd` ファイルを更新しません。

**pwdck** コマンドは、データベース管理セキュリティー・ファイル (`/etc/passwd.nm.idx`、`/etc/passwd.id.idx`、`/etc/security/passwd.idx`、および `/etc/security/lastlog.idx` ファイル) が最新の内容で、対応するシステム・セキュリティー・ファイルより新しいかどうかを検査します。`/etc/security/lastlog.idx` が `/etc/security/lastlog` より古くても問題はないことに注意してください。データベース管理セキュリティー・ファイルの内容が古ければ、警告メッセージが表示されて、root ユーザーが **mkpasswd** コマンドを実行しなければならないことを示します。

一般に、**sysck** コマンドは、トラステッド・システムの検査の一部として **pwdck** コマンドをコールします。また、root ユーザーまたはセキュリティー・グループのメンバーはこのコマンドを入力できます。

注: **auth2** 属性は推奨されないので、使用しないでください。

## フラグ

**-n** エラーを修正せずに報告します。  
**-p** エラーを報告せずに修正します。  
**-t** エラーを報告して、修正すべきかどうかを尋ねます。  
**-y** エラーを修正して報告します。

## セキュリティー

アクセス制御: このコマンドは、root ユーザーとセキュリティー・グループのメンバーに実行 (x) アクセス権を与えなければなりません。コマンドは、root ユーザーが認証情報の読み取りと書き込みができるように **setuid** し、**trusted computing base** 属性を持たなければなりません。

アクセスされるファイルは次のとおりです。

モード	ファイル
<b>rw</b>	<code>/etc/passwd</code>
<b>r</b>	<code>/etc/security/user</code>

モード	ファイル
rw	/etc/security/passwd
r	/etc/security/login.cfg

監査イベントは次のとおりです。

イベント	情報
PASSWORD_Check	ユーザー、エラー/修正、状況
PASSWORD_Ckerr	ファイル/ユーザー、エラー、状況

## 例

1. ローカル・ユーザー全員が有効なパスワードを持っているかどうかを検査するには、次のように入力します。

```
pwdck -y ALL
```

ここでは、エラーが報告されて解決されます。

2. ユーザー ariel が **/etc/security/passwd** ファイル内に有効なスタンザを持っているかどうかを検査するには、次のように入力します。

```
pwdck -y ariel
```

## ファイル

<b>/usr/bin/pwdck</b>	<b>pwdck</b> コマンドが入っています。
<b>/etc/passwd</b>	基本的なユーザー属性が入っています。
<b>/etc/security/passwd</b>	実際のパスワードおよびセキュリティ情報が入っています。
<b>/etc/security/user</b>	ユーザーの拡張属性が入っています。
<b>/etc/security/login.cfg</b>	構成情報とパスワード制約事項が入っています。

## 関連情報

**grpck** コマンド、 **mkpasswd** コマンド、 **sysck** コマンド、 **usrck** コマンド。

ユーザーの識別と認証、任意アクセス制御、トラステッド・コンピューティング・ベース、および監査については、セキュリティのネットワークの保護のセクションで説明しています。

---

## pwtokey コマンド

### 目的

パスワードを、ローカライズされた認証、ローカライズされない認証、および秘密鍵に変換します。

### 構文

```
pwtokey [-e ] [ -d DebugLevel ] [ -p Protocol ] [ -u KeyUsage ] [ -s ] Password [ EngineID | HostName | IPAddress ]
```

## 説明

AIX は **pwtokey** と呼ばれる機能を提供し、これによってパスワードを、ローカライズされた認証、ローカライズされない認証、および秘密鍵に変換できるようにします。 **pwtokey** プロシージャは、パスワードおよびエージェントの ID を入力とし、認証鍵と秘密鍵を生成します。 **pwtokey** 機能が使用するプロシージャは、 **clsnmp** コマンドが使用するアルゴリズムと同じなので、SNMP エージェントを構成する人は、特定のパスワードとエージェントが実行される IP アドレスを指定すれば、該当の認証鍵と秘密鍵を生成してユーザー用の **snmpd.conf** ファイルに入れることができます。

IP アドレスまたはホスト名を指定する場合、SNMP エージェントは AIX エージェントでなければなりません。 **engineID** は、エージェントの IP アドレスと AIX を表すエンタープライズ ID を使用する、ベンダー特有の公式を使用して作成されます。

## フラグ

**-d** *DebugLevel*

このフラグは、デバッグ情報の望ましいレベルを指定します。デバッグ・トレースは、オンまたはオフのいずれかであり、値 1 は、デバッグ・トレースをコマンド実行者の画面 (sysout) に生成し、値 0 はデバッグ・トレースを生成しないことを指定します。デフォルトでは、デバッグ・トレースはオフ (0) です。

**-e**

このフラグは、鍵を定義するエージェントを、IP アドレスまたはホスト名ではなく、**engineID** で識別することを示します。

**-p** *Protocol*

このフラグは、鍵を生成すべきプロトコルを指定します。有効な値は以下のとおりです。

### HMIC-MD5

HMIC-MD5 認証プロトコルで使用する鍵を生成する。

### HMIC-SHA

HMIC-SHA 認証プロトコルで使用する鍵を生成する。

**all** HMIC-MD5 および HMIC-SHA の両方の鍵を生成する。

デフォルトでは、HMIC-MD5 プロトコルの鍵が生成されます。

**-s**

このフラグは、読みやすくするために、スペースを追加して出力データを表示することを指定します。デフォルトでは、データはスペースのないフォーマットで表示されますが、その理由は、鍵を構成ファイルまたはコマンド行にカット・アンド・ペースト操作できるようにするためです。このフラグは、鍵に用途を示します。有効な値は以下のとおりです。

**-u** *KeyUsage*

**auth** 認証鍵。

**priv** 秘密鍵。

**all** 認証鍵および秘密鍵の両方。

注: 認証鍵と秘密鍵には、違いはありません。ただし、秘密鍵の長さは、その鍵をローカライズするかどうかにより異なります。



## パラメーター

<i>EngineID</i>	鍵を使用する SNMP エージェントの engineID を指定します。 engineID は、SNMP エージェントの初期設定時に snmpd.boots ファイルから決められます。 engineID は 1 から 32 のオクテット (2 から 64 の 16 進数) の文字列でなければなりません。デフォルトでは、エージェント ID は engineID ではありません。
<i>HostName</i>	SNMP 要求で鍵を使用する SNMP エージェントを指定します。
<i>IPAddress</i>	SNMP 要求でキーを使用する SNMP エージェントの IP アドレスを、ドット 10 進数表記で指定します。
<i>Password</i>	鍵の生成に使用するテキスト文字列を指定します。パスワードの長さは 8 から 255 文字の範囲でなければなりません。通常、パスワードには任意の印刷可能な文字を使用できますが、AIX シェルはある種の文字について、これを pwtokey コマンドに渡さずに、解釈します。AIX シェルによる文字の解釈を避けるには、パスワードを単一引用符で囲んでください。 <b>注:</b> このパスワードは、コミュニティー・ベースのセキュリティ (SNMPv1 および SNMPv2c) で使用されるコミュニティー名 (または "password") には関係しません。このパスワードは、まったく異なるセキュリティ機構である、ユーザー・ベースのセキュリティ用の鍵を生成するためにのみ使用されます。

## 例

1. この例は、pwtokey コマンドを単純に呼び出しています。

```
pwtokey testpassword 9.67.113.79
```

このコマンドの出力は以下のようなものです。

```
Display of 16 byte HMAC-MD5 authKey:  
775b109f79a6b71f94cca5d22451cc0e
```

```
Display of 16 byte HMAC-MD5 localized authKey:  
de25243d5c2765f0ce273e4bcf941701
```

この例が示すように、**pwtokey** は 2 つのキーを生成し、その 1 つはローカライズされ (指定されたエージェントでのみ使用できるように調整される)、もう 1 つはローカライズされません。通常、ローカライズされたキーが SNMP エージェントの構成で使用されます。ローカライズされないキーは、**clsnmp** コマンドの構成で使用されます。

2. 次の例では、**pwtokey** を呼び出して、認証とプライバシーの両方のための HMAC-SHA キーを要求しています。

```
pwtokey -p HMAC-SHA -u all testpassword 9.67.113.79
```

このコマンドの出力は以下のようなものです。

```
Display of 20 byte HMAC-SHA authKey:  
b267809aee4b8ef450a7872d6e348713f04b9c50
```

```
Display of 20 byte HMAC-SHA localized authKey:  
e5438092d1098a43e27e507e50d32c0edaa39b7c
```



```
Display of 20 byte HMAC-SHA privKey:  
b267809aee4b8ef450a7872d6e348713f04b9c50
```

```
Display of 16 byte HMAC-SHA localized privKey:  
e5438092d1098a43e27e507e50d32c0e
```

プライバシー・キーの出力は認証キーの出力と同じですが、ローカライズされたプライバシー・キーは、DES で要求された 16 バイトに切り捨てられている点が異なります。

注: 暗号化を使用する場合は、認証とプライバシーに異なるパスワードを使用する方がより安全です。

## 関連情報

**clsnmp** コマンド、**pwchange** コマンド、**snmpdv3** デーモン。

**/etc/clsnmp.conf** ファイル、**/etc/snmpdv3.conf** ファイル。

---

## pxed コマンド

### 目的

Preboot Execution Environment (PXE) プロキシ動的ホスト構成プロトコル (DHCP) サーバーをインプリメントします。

### 構文

システム・リソース・コントローラーを使用して **pxed** デーモンを開始するには、以下のようになります。

```
startsrc -s pxed [ -a]
```

システム・リソース・コントローラーを使用しないで **pxed** デーモンを開始するには、以下のようになります。

```
pxed [ -f ConfigurationFile]
```

### 説明

PXE は、プロトコルとメカニズムを定義して、ネットワーク接続のクライアント・システムがブート・イメージをネットワーク・サーバーから自動的にダウンロードし、オペレーティング・システムを始動できるようにします。BOOTP および DHCP プロトコルの拡張である PXE は、アドミニストレーター (必ずしも DHCP またはネットワーク・アドミニストレーターではない) が、PXE 対応クライアント・システムにインストールされたオペレーティング・システムを管理するための構成機能を提供します。

DHCP サーバーと同様、PXE Proxy DHCP サーバーは、PXE クライアントが自分の該当ブート・ファイルを見つけ出し、それをネットワーク・サーバーからダウンロードするために必要な情報を提供します。ただし、PXE Proxy DHCP サーバーは、クライアント IP アドレスや他の DHCP クライアント・オプションを管理しません。

PXE Proxy DHCP サーバーは、システム・ブート・イメージの管理と、DHCP アドレスと DHCP クライアント・ネットワーク構成の管理とを分離する必要があるときに使用するものです。**pxed** デーモンは、DHCP サーバーであるシステム上でも、DHCP サーバーでないシステム上でも実行するように構成できます。

## フラグ

<b>-a</b>	提供される引数。
<b>-f</b> <i>ConfigurationFile</i>	サーバーが使用する構成ファイルのパスと名前を指定します。これを指定しないと、 <b>/etc/pxed.cnf</b> がデフォルトになります。

## 終了状況

このコマンドは、以下の終了値を戻します。

0	正常終了。
>0	エラーが発生しました。

## セキュリティ

アクセス制御: このコマンドを実行するには、**root** 権限を持っていないければなりません。

PXE プロトコルは、非特権ユーザーを PXE クライアント・ブート・イメージのアドミニストレーターにすることができます。それは、**pxed** デーモンが、よく知られた保護 DHCP サーバー・ポート以外のポート上のクライアント・メッセージを **listen** するからです。ただし、このような環境を構成するには、DHCP サーバーを **pxed** デーモンと同じサーバー・システム上で実行し、**pxed** デーモンに関するファイル権限を非 **root** 実行用に変更する必要があります。

## ファイル

<b>/usr/sbin/pxed</b>	PXE Proxy DHCP サーバー・デーモンが含まれています。
<b>/usr/sbin/db_file.dhcpo</b>	PXE Proxy DHCP サーバーと DHCP サーバーが、構成情報の保管、検索、および管理のために使用するデータベースをインプリメントします。
<b>/etc/pxed.cnf</b>	<b>pxed</b> デーモンのデフォルトの構成ファイル。

## 関連情報

**dhcpsd** デーモン、 **binld** デーモン。

---

## qadm コマンド

### 目的

プリンター・スプーリング・システムのシステム管理機能を実行します。

### 構文

```
qadm { -G } | { [ -D Printer ] [ -K Printer ] [ -U Printer ] [ -X Printer ] }
```

### 説明

**qadm** コマンドは **enq** コマンドへのフロントエンド・コマンドです。このコマンドはプリンター、キュー、およびスプーリング・システムを接続または切断し、ジョブも取り消します。また、**qadm** コマンドは、要求されたフラグを **enq** コマンドが実行できるフォーマットに変換します。

**qadm** コマンドは、ローカル印刷ジョブについてのみ機能します。リモート印刷はサポートされません。

注: このコマンドを実行するには、root ユーザー権限があるか、printq グループに属していなければなりません。

Web-based System Manager (wsm) のプリンター・キュー・アプリケーションを使用してプリンター特性を変更できます。

また、System Management Interface Tool (SMIT) の **smit qadm** 高速パスを使用してこのコマンドを実行することもできます。

## フラグ

- D Printer** *Printer* 変数で指定したプリンターを切断します。 **qdaemon** プロセスはデバイスへのジョブの送信を停止します。 **qchk -P Printer** コマンド (この *Printer* は **-D** フラグ内の *Printer* 変数と同じ) を入力すると、デバイスの状況は *down* と報告されます。 **qadm** コマンドを使用すると、プリンターを停止する前に現行ジョブを終了できます。
- G** 非強制的にキューイング・システムを切断します。このフラグは、すべてのキューの現在実行中のジョブがすべて終了してから **qdaemon** プロセスを一時的に中断させます。このフラグを使わなければ、キュー内のジョブが停止するなどの問題を発生させずにシステムをダウンさせることはできません。
- K Printer** *Printer* 変数で指定したプリンターを切断します。すべての現行ジョブがただちに終了します。ジョブはそのままキュー内に残り、プリンターが元通り立ち上がると再び実行されます。
- U Printer** *Printer* 変数で指定したプリンターを切断します。 **qdaemon** プロセスは再びジョブをプリンターに送信します。 **qchk -P Printer** コマンド (この *Printer* は **-U** フラグの *Printer* 変数と同じ) を入力すると、デバイスの状況は *ready* として表示されます。
- X Printer** コマンドを実行したユーザーのジョブをすべて取り消します。 root ユーザー特権を持つユーザー、または printq グループのメンバーであるユーザーの場合は、そのキュー・システム上のすべてのジョブが取り消されます。

注: **-U** フラグと **-D** フラグがともに使用された場合には、**-U** フラグの方が優先順位が高くなります。

## 例

1. キューイング・システムを非強制的に切断するには、次のように入力します。

```
qadm -G
```

2. プリンター lp0 上の特定のユーザーのジョブをすべて取り消すか、root ユーザー権限を持っている場合はプリンター lp0 上のすべてのジョブを取り消すには、次のように入力します。

```
qadm -X lp0
```

3. キュー lp0 に接続されたプリンター lpd0 を接続するには、次のように入力します。

```
qadm -U lp0:lpd0
```

## ファイル

**/usr/sbin/qdaemon**  
**/var/spool/lpd/qdir/\***  
**/var/spool/lpd/stat\***

**qdaemon** デーモンが入っています。  
ジョブ記述ファイルが入っています。  
デバイスの状況に関する情報が入っています。

`/var/spool/qdaemon/*`  
`/etc/qconfig`  
`/etc/qconfig.bin`

キューに入れられたファイルの一時コピーが入っています。  
構成ファイルが入っています。  
`/etc/qconfig` ファイルの要約されたバイナリー・バージョンが入っています。

## 関連情報

`enq` コマンド、`qcan` コマンド、`qdaemon` コマンド、`qchk` コマンド、`qpri` コマンド、`qprt` コマンド。

`/etc/qconfig` ファイル。

プリンターおよび印刷 の 印刷キューの開始および停止 のセクション。

プリンターおよび印刷 の印刷スプーラーのセクション。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

---

## qcan コマンド

### 目的

印刷ジョブを取り消します。

### 構文

```
qcan [ -X ] [ -x JobNumber ] [ -P Printer ]
```

### 説明

`qcan` コマンドは特定のジョブ番号または印刷キュー内のすべてのジョブを取り消します。

Web-based System Manager (wsm) のプリンター・キュー・アプリケーションを使用してプリンター特性を変更できます。

System Management Interface Tool (SMIT) の `smit qcan` 高速パスを使用してこのコマンドを実行することもできます。

AIX 4.3.2 以降では、`qstatus` は、重複した 3 桁のジョブ番号を示して、ローカル・キューの管理を向上させるように拡張されています。状況コマンドの `enq`、`qchk`、`lpstat`、および `lpq` に `-W` フラグを指定すれば、より多くのジョブ番号桁を表示できます。

キューに重複している 3 桁のジョブ番号が表示されている場合は、`qchk -W` を使用すればより精度の高いジョブ番号がリストされます。その後で特定ジョブを取り消すことができます。

例えば、`qchk` を使用すると、ジョブ番号 123 を 2 回表示できますが、これに対し、`qchk -W` を使用した場合は、ジョブ番号の 1123 と 2123 が表示されます。ジョブ番号 2123 を取り消す場合は、`qcan -x 123` を指定すると、`qdaemon` によって、その内部リスト内で検出した最初の突き合わせジョブ番号が取り消されますが、このジョブ番号は、1123 である可能性があります。`-W` フラグが提供する追加情報を得ることで、特定のジョブ番号を取り消すことができます。

## フラグ

<b>-P</b> <i>Printer</i>	すべてのジョブまたは選択したジョブ番号を取り消す <i>Printer</i> を指定します。
<b>-x</b> <i>JobNumber</i>	<i>JobNumber</i> 変数で指定されたジョブ番号のみを取り消すことを指定します。
<b>-X</b>	すべてのジョブまたは指定されたプリンターに関するすべてのジョブを取り消します。 <i>root</i> ユーザー権限がある場合は、そのキュー上のすべてのジョブが削除されます。 <i>root</i> ユーザー権限がない場合は、ユーザーが実行要求したジョブのみが取り消されます。このフラグは、ローカル印刷ジョブに対してのみ有効です。

## 例

1. プリンター *lp0* 上でキューに入れられたすべてのジョブを取り消すには、次のように入力します。

```
qcan -X -P lp0
```

2. ジョブが送られるプリンターに関係なくジョブ番号 *123* を取り消すには、次のように入力します。

```
qcan -x 123
```

## ファイル

<i>/usr/sbin/qdaemon</i>	<i>qdaemon</i> デーモンが入っています。
<i>/var/spool/lpd/qdir/*</i>	ジョブ記述ファイルが入っています。
<i>/var/spool/lpd/stat/*</i>	デバイスの状況に関する情報が入っています。
<i>/var/spool/qdaemon/*</i>	キューに入れられたファイルの一時コピーが入っています。
<i>/etc/qconfig</i>	構成ファイルが入っています。
<i>/etc/qconfig.bin</i>	<i>/etc/qconfig</i> ファイルの要約されたバイナリー・バージョンが入っています。

## 関連情報

*enq* コマンド、 *qadm* コマンド、 *qchk* コマンド、 *qpri* コマンド、 *qprt* コマンド。

*/etc/qconfig* ファイル。

オペレーティング・システムおよびデバイスのマネージの印刷ジョブを取り消す方法 (*qcan* コマンド) のセクション。

プリンターおよび印刷 の 印刷スプーラー のセクション。

Web-based System Manager のインストールについては、 *AIX 5L Version 5.3 Web-based System Manager 管理ガイド* の第 2 章 インストールおよびシステム要件のセクションを参照してください。

---

## qchk コマンド

### 目的

印刷キューの状況を表示します。

### 構文

```
qchk [ -A ] [ -L | -W ] [ -P Printer ] [ -# JobNumber ] [ -q ] [ -u UserName ] [ -w Delay ]
```

## 説明

**qchk** コマンドは、指定された印刷ジョブ、印刷キュー、またはユーザーに関する現在の状況情報を表示します。特定の状況情報を表示するには、適切なフラグの後に要求したい名前または番号を指定します。フラグを指定せずに **qchk** コマンドを実行すると、デフォルト・キューの状況が戻されます。

Web-based System Manager (wsm) のプリンター・キュー・アプリケーションを使用してプリンター特性を変更できます。

また、System Management Interface Tool (SMIT) の **smit qchk** 高速パスを使用してこのコマンドを実行することもできます。

## フラグ

**-#JobNumber** *JobNumber* 変数で指定されたジョブ番号の状況を要求します。 **-#JobNumber** フラグが単独で使用されているときには、**qchk** コマンドはデフォルト・キュー上で *JobNumber* を検索します。すべてのキューで *JobNumber* を検索するには、**-#** フラグを **-A** フラグとともに使用しなければなりません。また、**-#** フラグを **-P Queue** フラグとともに使用することもできます。

### 注:

1. デフォルトの宛先プリンターをオーバーライドするには、**-P Queue** を指定します。
2. ジョブ 1、2、および 3 がプリンター・キューに入っている場合、ジョブ 1 の実行時にジョブ 3 の状況を表示するように指定しても、状況情報はジョブ 3 だけでなく、ジョブ 1 とジョブ 3 を表示します。
3. 存在しないジョブ番号を指定すると、システムはエラー・メッセージの代わりに、キュー上の現行ジョブ番号を表示します。

**-A** すべてのキューの状況を要求します。

**-L** 情報を長形式モードで表示することを指定します。このフラグは **-W** フラグとともに使用することはできません。**-L** フラグと **-W** フラグを同時に使用する場合、最初に指定されたフラグが優先されます。

**-P Printer** *Printer* 変数で指定されたプリンターの状況を要求します。

**-q** デフォルト印刷キューの状況を要求します。

**-u UserName** *UserName* 変数で指定されたユーザーが送信したすべての印刷ジョブの状況を要求します。

**-W** これを指定すると、さらに長いキュー名、デバイス名、およびジョブ番号の広幅フォーマットのモードで情報が表示されます。より大きいジョブ番号情報は、AIX バージョン 4.3.2 以降で提供されます。このフラグは **-L** フラグとともに使用することはできません。**-L** フラグと **-W** フラグを同時に使用する場合、最初に指定されたフラグが優先されます。

**-w Delay** すべての印刷ジョブが終了するまで、要求された状況情報を *Delay* 変数によって秒単位で指定された間隔で更新します。

## 例

1. デフォルトの印刷キューを表示するには、次のように入力します。

```
qchk -q
```

2. 空になるまで 5 秒ごとに画面を更新しながら、すべてのキューの長時間状況を表示するには、次のように入力します。

```
qchk -A -L -w 5
```

3. プリンター lp0 の状況を表示するには、次のように入力します。

```
qchk -P lp0
```

- ジョブ番号 123 の状況を表示するには、次のように入力します。

```
qchk -# 123
```

- キューの状況をプリンター lp0 に限定して、すべての印刷ジョブの状況を表示するには、次のように入力します。

```
qchk -A -P lp0
```

- AIX 4.2.1 またはそれ以降のバージョンのデフォルトの印刷キューを広フォーマットで表示するには、次のように入力します。

```
qchk -W -q
```

## ファイル

<code>/usr/sbin/qdaemon</code>	<code>qdaemon</code> デーモンが入っています。
<code>/var/spool/lpd/qdir/*</code>	ジョブ記述ファイルが入っています。
<code>/var/spool/lpd/stat/*</code>	デバイスの状況に関する情報が入っています。
<code>/var/spool/qdaemon/*</code>	キューに入れられたファイルの一時コピーが入っています。
<code>/etc/qconfig</code>	構成ファイルが入っています。
<code>/etc/qconfig.bin</code>	<code>/etc/qconfig</code> ファイルの要約されたバイナリー・バージョンが入っています。

## 関連情報

プリンターおよび印刷 の 印刷スプーラー のセクション。

`enq` コマンド、 `qadm` コマンド、 `qcan` コマンド、 `qpri` コマンド、 `qprt` コマンド。

`/etc/qconfig` ファイル。

Web-based System Manager のインストールについては、 *AIX 5L Version 5.3 Web-based System Manager 管理ガイド* の第 2 章 インストールおよびシステム要件のセクションを参照してください。

オペレーティング・システムおよびデバイスのマネージの印刷ジョブの状況を検査するコマンド (`qchk` コマンド) のセクション。

---

## qdaemon コマンド

### 目的

`enq` コマンドでキューに入れられたジョブをスケジュールします。

### 構文

```
qdaemon
```

### 説明

`qdaemon` コマンドは、 `enq` コマンドがキューに入れられたジョブの印刷スケジュールを行うバックグラウンド・プロセスです (通常は `startsrc` コマンドで始動されます)。



推奨: `/etc/qconfig` ファイルを編集するには、`chque`、`mkque`、`rmque`、`chqueuedev`、`mkqueuedev`、および `rmqueuedev` の各コマンドまたは `SMIT` を使用します。さらに、これらのコマンドを実行する時間帯は、ピークが弱まった時、またはオフピーク時にすることをお勧めします。

手動操作による `/etc/qconfig` ファイルの編集が必要な場合、まず `enq -G` コマンドを発行し、すべてのジョブがプロセスされた後、キューイング・システムと `qdaemon` を一時停止状態にしてください。次に、`/etc/qconfig` ファイルを編集し、新しい構成を使用して `qdaemon` を再始動できます。

## セキュリティ

特権制御: `root` ユーザーと `printq` グループのメンバーだけが、このコマンドへの実行 (x) アクセス権を持つものとしてます。

監査イベント  
`ENQUE_exec`

情報  
キュー名、ジョブ名、ホスト名、ファイル名、ユーザー名

## ファイル

<code>/usr/sbin/qdaemon</code>	<code>qdaemon</code> デーモンが入っています。
<code>/var/spool/lpd/qdir/*</code>	ジョブ記述ファイルが入っています。
<code>/var/spool/lpd/pio/@local/fullmsg</code>	完全な情報を含めるために <code>qdaemon</code> メッセージを活動化するフラグ・ファイルが入っています。
<code>/var/spool/lpd/stat/*</code>	デバイスの状況に関する情報が入っています。
<code>/var/spool/qdaemon/*</code>	キューに入れられたファイルの一時コピーが入っています。
<code>/etc/qconfig</code>	構成ファイルが入っています。
<code>/etc/qconfig.bin</code>	<code>/etc/qconfig</code> ファイルの要約されたバイナリー・バージョンが入っています。

## 関連情報

`cancel` コマンド、`chque` コマンド、`chqueuedev` コマンド、`disable` コマンド、`enable` コマンド、`qstatus` コマンド、`lp` コマンド、`lpd` コマンド、`lpq` コマンド、`lpr` コマンド、`lprm` コマンド、`lpstat` コマンド、`lsallq` コマンド、`lsallqdev` コマンド、`lsque` コマンド、`lsqueuedev` コマンド、`mkque` コマンド、`mkqueuedev` コマンド、`qprt` コマンド、`qadm` コマンド、`qchk` コマンド、`qcan` コマンド、`qpri` コマンド、`rmque` コマンド、`rmqueuedev` コマンド、`startsrc` コマンド。

`/etc/qconfig` ファイル。

プリンターおよび印刷 の 印刷の管理のセクション。

プリンターおよび印刷 の 印刷スプーラーのセクション。

*AIX 5L* バージョン 5.3 ナショナル・ランゲージ・サポート ガイドおよびリファレンスのナショナル・ランゲージ・サポートの概要のセクション。

オペレーティング・システムおよびデバイスのマネージのシステム・リソース・コントローラーのセクション。

プリンターおよび印刷 の バックエンドおよび `qdaemon` の相互作用のセクション。

---

## qhld コマンド

### 目的

スプールされた印刷ジョブを保持し、解放します。

### 構文

```
qhld [ -r ] { -#JobNumber [ -PQueue ] | -PQueue | -uUser [ -PQueue ] }
```

### 説明

**qhld** コマンドは、印刷ジョブをスプール状態で保持します。保持するジョブは、ジョブ番号、キュー名、またはユーザー名で指定します。**-r** フラグを指定すると、印刷ジョブの保持が解除されます。

AIX 4.3.2 以降では、**qstatus** は、重複した 3 桁のジョブ番号を示して、ローカル・キューの管理を向上させるように拡張されています。状況コマンドの **enq**、**qchk**、**lpstat**、および **lpq** に **-W** フラグを指定すれば、より多くのジョブ番号桁を表示できます。

キューに重複している 3 桁のジョブ番号が表示されている場合は、**qchk -W** を使用すればより精度の高いジョブ番号がリストされます。その後で特定ジョブを保持することができます。

例えば、**qchk** を使用すると、ジョブ番号 123 を 2 回表示できますが、これに対し、**qchk -W** を使用した場合は、ジョブ番号の 1123 と 2123 が表示されます。ジョブ番号 2123 を保持する場合は、**qhld -# 123** を指定すると、**qdaemon** によって、その内部リスト内で検出した最初の突き合わせジョブ番号が保留されますが、このジョブ番号は、1123 である可能性があります。**qstatus -W** フラグが提供する追加情報を得ることで、特定のジョブ番号を保持することができます。

### フラグ

<b>-#JobNumber</b>	保持する印刷ジョブの番号を指定します。
<b>-P Queue</b>	保持する印刷キューを指定します。
<b>-r</b>	番号、キュー、またはユーザー名で印刷ジョブを解放します。
<b>-uUser</b>	保持する印刷ジョブが属するユーザーの名前を指定します。

### 例

- 印刷ジョブ番号 300 を保持するには、次のように入力します。

```
qhld -#300
```
- キュー lp0 のすべての印刷ジョブを保持するには、次のように入力します。

```
qhld -P lp0
```
- ユーザー fred のすべてのジョブを保持するには、次のように入力します。

```
qhld -u fred
```
- ジョブ番号 300 を解放するには、次のように入力します。

```
qhld -#300 -r
```
- キュー lp0 のすべてのジョブを解放するには、次のように入力します。

```
qhld -Plp0 -r
```
- ユーザー fred のすべてのジョブを解放するには、次のように入力します。

```
qhld -u fred -r
```

## ファイル

<code>/usr/sbin/qdaemon</code>	<code>qdaemon</code> デーモンが入っています。
<code>/var/spool/lpd/qdir/*</code>	ジョブ記述ファイルが入っています。
<code>/var/spool/lpd/stat/*</code>	デバイスの状況に関する情報が入っています。
<code>/var/spool/qdaemon/*</code>	キューに入れられたファイルの一時コピーが入っています。
<code>/etc/qconfig</code>	構成ファイルが入っています。
<code>/etc/qconfig.bin</code>	<code>/etc/qconfig</code> ファイルの要約されたバイナリー・バージョンが入っています。

## 関連情報

`qprt` コマンド、`qmov` コマンド、`enq` コマンド。

プリンターおよび印刷 の 印刷の管理 のセクション。

プリンターおよび印刷 の 印刷スプーラー のセクション。

---

## qmov コマンド

### 目的

スプールされた印刷ジョブを別のキューに移動します。

### 構文

```
qmov -mNewQueue { -#JobNumber [ -PQueue ] | -PQueue | -uUser [ -PQueue ] }
```

### 説明

`qmov` コマンドは、スプールされた印刷ジョブを別の印刷キューに移動します。移動する印刷ジョブは、ジョブ番号、キュー、またはユーザー名で識別します。このコマンドのフォーマットは、ジョブの移動先となるキューを第 1 引数、移動するジョブの名前を第 2 引数として指定する必要があります。

AIX 4.3.2 以降では、`qstatus` は、重複した 3 桁のジョブ番号を示して、ローカル・キューの管理を向上させるように拡張されています。状況コマンドの `enq`、`qchk`、`lpstat`、および `lpq` に `-W` フラグを指定すれば、より多くのジョブ番号桁を表示できます。

キューに重複している 3 桁のジョブ番号が表示されている場合は、`qchk -W` を使用すればより精度の高いジョブ番号がリストされます。その後で特定ジョブを移動できます。

例えば、`qchk` を使用すると、ジョブ番号 123 を 2 回表示できますが、これに対し、`qchk -W` を使用した場合は、ジョブ番号の 1123 と 2123 が表示されます。ジョブ番号 2123 を移動する場合は、`qmov -# 123` を指定すると、`qdaemon` によって、その内部リスト内で検出した最初の突き合わせジョブ番号が移動されますが、このジョブ番号は、1123 である可能性があります。`qstatus -W` フラグが提供する追加情報を得ることで、特定のジョブ番号を移動できます。

## フラグ

<code>-#JobNumber</code>	移動する印刷ジョブのジョブ番号を指定します。
<code>-mNewQueue</code>	宛先の印刷キューの名前を指定します。
<code>-P Queue</code>	移動するジョブの現在の印刷キューを指定します。
<code>-uUser</code>	移動する印刷ジョブが属するユーザーの名前を指定します。

## 例

1. ジョブ番号 280 をキュー 1p0 に移動するには、次のように入力します。

```
qmov -m1p0 -#280
```

2. キュー 1p1 のすべての印刷ジョブをキュー 1p0 に移動するには、次のように入力します。

```
qmov -m1p0 -P1p1
```

3. Mary のすべての印刷ジョブをキュー 1p0 に移動するには、次のように入力します。

```
qmov -m1p0 -u mary
```

## ファイル

**/usr/sbin/qdaemon**  
**/var/spool/lpd/qdir/\***  
**/var/spool/lpd/stat\***  
**/var/spool/qdaemon/\***  
**/etc/qconfig**  
**/etc/qconfig.bin**

**qdaemon** デーモンが入っています。  
ジョブ記述ファイルが入っています。  
デバイスの状況に関する情報が入っています。  
キューに入れられたファイルの一時コピーが入っています。  
構成ファイルが入っています。  
**/etc/qconfig** ファイルの要約されたバイナリー・バージョンが入っています。

## 関連情報

**enq** コマンド、 **qhld** コマンド、 **qprt** コマンド。

プリンターおよび印刷 の 印刷の管理 のセクション。

プリンターおよび印刷の Print spooler のセクション。

---

## qosadd コマンド

### 目的

QoS (Quality of Service) Service Category または Policy Rule を追加します。

### 構文

Service Category を追加するには、次のように入力します。

```
qosadd [ -s ServiceCategory] [ -t OutgoingTOS] [ -b MaxTokenBucket] [ -f FlowServiceType] [ -m MaxRate] service
```

Policy Rule を追加するには、次のように入力します。

```
qosadd [ -s ServiceCategory] [ -r ServicePolicyRules] [ -l PolicyRulePriority] [ -n ProtocolNumber] [ -A SrcAddrRange] [ -a DestAddrRange] [ -P SrcPortRange] [ -p DestPortRange] policy
```

### 説明

**qosadd** コマンドは、**policyd.conf** ファイル内の指定された Service Category または Policy Rule エントリーを追加し、変更結果を QoS Manager にインストールします。

## フラグ

サービス追加のフラグは、以下のとおりです。

<b>-s</b>	<b>ServiceCategory</b> 属性の名前 (必須)。
<b>-t</b>	<b>OutgoingTOS</b> 属性。8 ビットの 2 進数で指定。
<b>-b</b>	<b>MaxTokenBucket</b> 属性。KB (k ビット) で指定。
<b>-f</b>	<b>FlowServiceType</b> 属性 (ControlledLoad または Guaranteed)。
<b>-m</b>	<b>MaxRate</b> 属性。Kbps (秒当たりのキロバイト数) で指定。

ポリシー追加のフラグは、以下のとおりです。

<b>-s</b>	<b>ServiceCategory</b> 属性の名前 (必須)。
<b>-r</b>	<b>ServicePolicyRules</b> 属性の名前 (必須)。
<b>-l</b>	<b>PolicyRulePriority</b> 属性 (正整数)。
<b>-n</b>	<b>ProtocolNumber</b> 属性。 <b>/etc/protocols</b> ファイルに定義済み。
<b>-A</b>	<b>SrcAddrRange</b> 属性。 a1 から a2 (a2 >= a1) の Source IP アドレス範囲。
<b>-a</b>	<b>DestAddrRange</b> 属性。 i1 から i2 (i2 >= i1) の Destination IP アドレス範囲。
<b>-P</b>	<b>SrcPortRange</b> 属性。 a1 から a2 (a2 >= a1) の Source Port 範囲。
<b>-p</b>	<b>DestPortRange</b> 属性。 i1 から i2 (i2 >= i1) の Destination Port 範囲。

## 終了状況

0	正常終了。
正整数	エラーが発生しました。

## 例

1. sc01 サービスを追加するには、次のように入力します。

```
qosadd -s sc01 -t 10000001 -b 81 -f ControlledLoad -m 41 service
```

2. pr01 ポリシーを追加するには、次のように入力します。

```
qosadd -s sc01 -r pr01 -l 2 -n 17 -A 9.3.25.1-9.3.25.10 -a 9.3.25.33-9.3.25.33 -p 9001-9010 -P 9000-9000 policy
```

3. sc02 サービスを追加するには、次のように入力します。

```
qosadd -s sc02 -t 10000001 -b 81 service
```

4. pr02 ポリシーを追加するには、次のように入力します。

```
qosadd -s sc02 -r pr02 -l 2 -n 17 policy
```

## 関連情報

**qosstat** コマンド、 **qosmod** コマンド、 **qosremove** コマンド、 **qoslist** コマンド。

---

## qoslist コマンド

### 目的

特定の QoS (Quality of Service) Service Category または Policy Rule、あるいはそれらのすべてをリストします。

## 構文

Service Category をリストするには、次のように入力します。

```
qoslist [ServiceCategory] service
```

Policy Rule をリストするには、次のように入力します。

```
qoslist [ServicePolicyRule] policy
```

## 説明

**qoslist** コマンドは、指定された Service Category または Policy Rule をリストします。特定の名前が指定されていない場合は、**qoslist** コマンドは、すべての Service Category または Policy Rule をリストします。

## 終了状況

0	正常終了。
正整数	エラーが発生しました。

## 例

1. sc01 サービスをリストするには、次のように入力します。  

```
qoslist sc01 service
```
2. pr01 ポリシーをリストするには、次のように入力します。  

```
qoslist pr01 policy
```
3. すべての QoS ServiceCategory をリストするには、次のように入力します。  

```
qoslist service
```
4. すべての QoS PolicyRule をリストするには、次のように入力します。  

```
qoslist policy
```

## 関連情報

**qosstat** コマンド、 **qosmod** コマンド、 **qosremove** コマンド、 **qosadd** コマンド。

---

## qosmod コマンド

### 目的

既存の QoS (Quality of Service) Service Category または Policy Rule を変更します。

### 構文

既存の Service Category を変更するには、次のように入力します。

```
qosmod [ -s ServiceCategory] [ -t OutgoingTOS] [ -b MaxTokenBucket] [ -f FlowServiceType] [ -m MaxRate] service
```

既存の Policy Rule を変更するには、次のように入力します。

**qosmod** [ **-s** *ServiceCategory*] [ **-r** *ServicePolicyRules*] [ **-l** *PolicyRulePriority*] [ **-n** *ProtocolNumber*] [ **-A** *SrcAddrRange*] [ **-a** *DestAddrRange*] [ **-P** *SrcPortRange*] [ **-p** *DestPortRange*] **policy**

## 説明

**qosmod** コマンドは、**policyd.conf** ファイル内の指定された Service Category または Policy Rule エントリーを変更し、変更結果を QoS Manager にインストールします。

**qosmod** コマンドは、古いポリシーの統計をすべてクリアします。**qosstat** コマンドを **qosmod** の直後に実行すると、古いルールを使用しており、変更されたルールに移ったすべてのデータ接続を、ユーザーは見るできません。これは、その接続にデータ・パケットが到着するまで、データ接続の再分類が遅れるためです。

注: ルールの優先順位またはフィルター仕様を変更すると、その特定のルールを使用するデータ接続が再分類されるだけです。他のルールを使用する接続は、既存の分類種別を維持します。

## フラグ

サービス変更のフラグは、以下のとおりです。

<b>-s</b>	<b>ServiceCategory</b> 属性の名前 (必須)。
<b>-t</b>	<b>OutgoingTOS</b> 属性。8 ビットの 2 進数で指定。
<b>-b</b>	<b>MaxTokenBucket</b> 属性。KB (キロバイト) で指定。
<b>-f</b>	<b>FlowServiceType</b> 属性 (ControlledLoad または Guaranteed)。
<b>-m</b>	<b>MaxRate</b> 属性。Kbps (秒当たりのキロバイト数) で指定。

ポリシー変更のフラグは、以下のとおりです。

<b>-s</b>	<b>ServiceCategory</b> 属性の名前 (必須)。
<b>-r</b>	<b>ServicePolicyRules</b> 属性の名前 (必須)。
<b>-l</b>	<b>PolicyRulePriority</b> 属性 (正整数)。
<b>-n</b>	<b>ProtocolNumber</b> 属性。 <b>/etc/protocols</b> ファイルに定義済み。
<b>-A</b>	<b>SrcAddrRange</b> 属性。a1 から a2 (a2 >= a1) の Source IP アドレス範囲。
<b>-a</b>	<b>DestAddrRange</b> 属性。i1 から i2 (i2 >= i1) の Destination IP アドレス範囲。
<b>-P</b>	<b>SrcPortRange</b> 属性。a1 から a2 (a2 >= a1) の Source Port 範囲。
<b>-p</b>	<b>DestPortRange</b> 属性。i1 から i2 (i2 >= i1) の Destination Port 範囲。

## 終了状況

0	正常終了。
正整数	エラーが発生しました。

## 例

1. sc01 サービスを変更するには、次のように入力します。

```
qosmod -s sc01 -t 10001100 -b 84 -f Guaranteed service
```

2. pr01 ポリシーを変更するには、次のように入力します。

```
qos -s sc01 -r pr01 -l 10 -n 6 -A 9.3.25.15-9.3.25.20 -a 9.3.25.39-9.3.25.39 -p 9015-9020 policy
```

3. sc02 サービスを変更するには、次のように入力します。

```
qosmod -s sc02 -t 10001111 service
```



4. pr02 ポリシーを変更するには、次のように入力します。

```
qosmod -s sc02 -r pr02 -l 13 -n 6 policy
```

## 関連情報

**qosstat** コマンド、 **qoslist** コマンド、 **qosremove** コマンド、 **qosadd** コマンド。

---

## qosremove コマンド

### 目的

QoS (Quality of Service) Service Category または Policy Rule を除去します。

### 構文

Service Category を除去するには、次のように入力します。

```
qosremove [ServiceCategory] service
```

Policy Rule を除去するには、次のように入力します。

```
qosremove [ServicePolicyRule] policy
```

カーネルにインストールされたすべての Policies および Service カテゴリを除去する場合:

```
qosremove all
```

### 説明

**qosremove** コマンドは、**policyd.conf** ファイル内の指定された Service Category または Policy Rule エントリー、および QoS Manager 内の関連するポリシーまたはサービスを除去します。

### 終了状況

0	正常終了。
正整数	エラーが発生しました。

### 例

1. sc01 サービスを除去するには、次のように入力します。

```
qosremove sc01 service
```

2. pr01 ポリシーを除去するには、次のように入力します。

```
qosremove pr01 policy
```

## 関連情報

**qosstat** コマンド、 **qosmod** コマンド、 **qoslist** コマンド、 **qosadd** コマンド。

---

## qosstat コマンド

### 目的

サービス品質 (QoS) 状況を表示します。

### 構文

```
qosstat [ -A ] [ -F ] [ -S ]
```

### 説明

**qosstat** コマンドは、インストールされた QoS ポリシーに関する情報を表示します。フラグを指定せずに **qosstat** を使用すると、インストールされたポリシーごとにフィルター/フロー仕様および統計情報が戻されます。

### フラグ

- A** インストールされたポリシーごとにポリシー・ルール・ハンドルを戻します。固有のハンドルが、インストールされたポリシーごとに qoS マネージャーによって割り当てられます。
- F** インストールされたポリシーごとにフローおよびフィルター仕様を戻します。
- S** インストールされたポリシーごとに統計情報を戻します。

### 例

#### 1. qosstat

```
Policy Rule handle 1:

Filter specification for rule index 1:
  PolicyRulePriority:          0
  protocol:                   TCP
  source IP addr:              INADDR_ANY
  destination IP addr:        INADDR_ANY
  source port:                 80
  destination port:           ANY_PORT

Flow Class for rule index 1:
  service class:              Diff-Serv
  peak rate:                  100000000 bytes/sec
  average rate:               128 bytes/sec
  bucket depth:               4096 bytes
  TOS (in profile):          0
  TOS (out profile):          0

Statistics for rule index 1:
  total number of connections: 0
  total bytes transmitted:     0
  total packets transmitted:   0
  total in-profile bytes transmitted: 0
  total in-profile packets transmitted: 0

Policy Rule Handle 2:

Filter specification for rule index 2:
  PolicyRulePriority:          0
  protocol:                   TCP
  source IP addr:              INADDR_ANY
  destination IP addr:        INADDR_ANY
  source port:                 100
  destination port:           ANY_PORT

Flow Class for rule index 2:
  service class:              Diff-Serv
```

```
peak rate:      100000000 bytes/sec
average rate:   128 bytes/sec
bucket depth:   4096 bytes
TOS (in profile): 0
TOS (out profile): 0
```

```
Statistics for rule index 2:
total number of connections:      0
total bytes transmitted:          0
total packets transmitted:        0
total in-profile bytes transmitted: 0
total in-profile packets transmitted: 0
```

## 2. qosstat -A

```
Policy Rule Handle 1:
rule index:      1
```

```
Policy Rule Handle 2:
rule index:      2
```

## 3. qosstat -F

```
Policy Rule Handle 1:
Filter specification for rule index 1:
PolicyRulePriority:      0
protocol:                TCP
source IP addr:          INADDR_ANY
destination IP addr:     INADDR_ANY
source port:             80
destination port:        ANY_PORT
```

```
Flow Class for rule index 1:
service class:           Diff-Serv
peak rate:              100000000 bytes/sec
average rate:           128 bytes/sec
bucket depth:           4096 bytes
TOS (in profile):      0
TOS (out profile):     0
```

```
Policy Rule Handle 2:
Filter specification for rule index 2:
PolicyRulePriority:      0
protocol:                TCP
source IP addr:          INADDR_ANY
destination IP addr:     INADDR_ANY
source port:             100
destination port:        ANY_PORT
```

```
Flow Class for rule index 2:
service class:           Diff-Serv
peak rate:              100000000 bytes/sec
average rate:           128 bytes/sec
bucket depth:           4096 bytes
TOS (in profile):      0
TOS (out profile):     0
```

## 4. qosstat -S

```
Statistics for rule index 1:
total number of connections:      0
total bytes transmitted:          0
total packets transmitted:        0
total in-profile bytes transmitted: 0
total in-profile packets transmitted: 0
```

```
Policy Rule Handle 2:
Statistics for rule index 2:
total number of connections:      0
total bytes transmitted:          0
total packets transmitted:        0
total in-profile bytes transmitted: 0
total in-profile packets transmitted: 0
```

## 関連情報

ネットワークおよびコミュニケーションのマネージ の TCP/IP のサービス品質 (QoS) のセクション。

---

## qpri コマンド

### 目的

印刷キュー内のジョブに優先順位を付けます。

### 構文

```
qpri -# JobNumber -a PriorityNumber
```

### 説明

**qpri** コマンドは、ジョブ番号を指定してそれに優先順位番号を割り当てることによって、印刷キューに入っているジョブの優先順位付けを行います。

**qpri** コマンドは、ローカル印刷ジョブおよびリモート・キューのローカル側に対してのみ機能します。リモート印刷ジョブはサポートされません。また、このコマンドを実行するには、root ユーザー権限があるか、printq グループに属していなければなりません。

Web-based System Manager (wsm) のプリンター・キュー・アプリケーションを使用してプリンター特性を変更できます。

System Management Interface Tool (SMIT) の **smit qpri** 高速パスを使用してこのコマンドを実行することもできます。

AIX 4.3.2 以降では、**qstatus** は、重複した 3 桁のジョブ番号を示して、ローカル・キューの管理を向上させるように拡張されています。状況コマンドの **enq**、**qchk**、**lpstat**、および **lpq** に **-W** フラグを指定すれば、より多くのジョブ番号桁を表示できます。

キューに重複している 3 桁のジョブ番号が表示されている場合は、**qchk -W** を使用すればより精度の高いジョブ番号がリストされます。その後で特定ジョブの優先順位を変更できます。

例えば、**qchk** を使用すると、ジョブ番号 123 を 2 回表示できますが、これに対し、**qchk -W** を使用した場合は、ジョブ番号の 1123 と 2123 が表示されます。ジョブ番号 2123 の優先順位を変更する場合は、**qpri -# 123** を指定すると、**qdaemon** によって、その内部リスト内で検出した最初の突き合わせジョブ番号の優先順位が変更されますが、このジョブ番号は、1123 である可能性があります。**qstatus -W** フラグが提供する追加情報を得ることで、特定のジョブ番号の優先順位を変更できます。

## フラグ

**-#JobNumber**

優先順位を変更するジョブの番号を指定します。

**-a PriorityNumber**

**JobNumber** 変数で指定された印刷ジョブに、新しい優先順位番号を指定します。優先順位番号の範囲は 1 から 20 までですが、root ユーザーまたは printq グループのメンバーの場合は、1 から 30 までの優先順位番号を選択することができます。

## 例

ジョブ番号 123 の優先順位番号を 18 に変更するには、次のように入力します。

```
qpri -# 123 -a 18
```

## ファイル

<code>/usr/sbin/qdaemon</code>	<code>qdaemon</code> デーモンが入っています。
<code>/var/spool/lpd/qdir</code>	ジョブ記述ファイルが入っています。
<code>/var/spool/lpd/stat</code>	デバイスの状況に関する情報が入っています。
<code>/var/spool/qdaemon</code>	キューに入れられたファイルの一時コピーが入っています。
<code>/etc/qconfig</code>	構成ファイルが入っています。
<code>/etc/qconfig.bin</code>	<code>/etc/qconfig</code> ファイルの要約されたバイナリー・バージョンが入っています。

## 関連情報

`enq` コマンド、`qadm` コマンド、`qcan` コマンド、`qchk` コマンド、`qpri` コマンド。

`/etc/qconfig` ファイル。

オペレーティング・システムおよびデバイスのマネージの印刷ジョブに優先順位を付けるコマンド (`qpri` コマンド) のセクション。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

---

## qpri コマンド

### 目的

印刷ジョブを開始します。

### 構文

```
qpri [ -a PreviewOption ] [ -A Level ] [ -b BottomMargin ] [ -B Value ] [ -c ] [ -C ] [ -d  
InputDataType ] [ -D "User" ] [ -e EmphasizedOpt ] [ -E DblHigh ] [ -f Filter ] [ -F Name ] [ -g Begin ]  
[ -G Coord ] [ -h "Header" ] [ -H "HostName" ] [ -i Indent ] [ -l FontPath ] [ -j Init ] [ -J Restore ] [ -k  
Color ] [ -K Condense ] [ -l Length ] [ -L LineWrap ] [ -m Message ] [ -M MessageFile ] [ -n ] [ -N  
NumberCopies ] [ -O PaperHand ] [ -p Pitch ] [ -P Queue [ :QueueDevice ] ] [ -Q Value ] [ -q Quality ]  
[ -r ] [ -R Priority ] [ -s NameType ] [ -S Speed ] [ -t TopMargin ] [ -T "Title" ] [ -u PaperSrc ] [ -U  
Directional ] [ -v LinesPerIn ] [ -V Vertical ] [ -w PageWidth ] [ -W DblWide ] [ -x LineFeed ] [ -X  
CodePage ] [ -y DblStrike ] [ -Y Duplex ] [ -z Rotate ] [ -Z FormFeed ] [ -# { j | h | v } ] [ --  
OutputBin ]{ File | - } ...
```

### 説明

`qpri` コマンドは、印刷ジョブを作成し、これをキューに入れて `File` パラメーターで指定されたファイルを印刷します。標準入力からのファイルを印刷するには、ファイル名の代わりに `-` (ダッシュ) を指定します。複数のファイルを指定すると、複数ファイルで 1 つの印刷ジョブが作成されます。`qpri` コマンドは、指定された順序でファイルを印刷します。

ファイルを印刷するには、このファイルに対する読み取りアクセス権を持っていないとなりません。 **-r** フラグを使用すると、ファイルを印刷した後で取り除くことができます。ファイルを除去するには、そのファイルが入っているディレクトリーに対する書き込みアクセス権を持っていないとなりません。 **qprt** コマンドに印刷ジョブ完了を通知させたい場合は、**-n** フラグを指定します。

**-B** フラグを **-D**、**-H**、および **-T** フラグとともに使用すると、バースト・ページをカスタマイズすることができます。バースト・ページは、印刷ジョブの先頭、終わり、またはその両方にマークを付けます。印刷ジョブの先頭と終わりにバースト・ページでマークを付けるには、**-B aa** フラグを使用します。

すべてのフラグはオプションであり、どのような順序で指定してもかまいません。 **qprt** コマンドは、フラグとその引数との間のスペースを無視します。 **-** (ダッシュ) の後に引数を持たないフラグをグループ化することができます。すべてのフラグとその引数を指定してから、*File* パラメーターを指定しなければなりません。

Web-based System Manager (wsm) のプリンター・キュー・アプリケーションを使用してプリンター特性を変更できます。

このコマンドを実行するときに、System Management Interface Tool (SMIT) の **smit qprt** 高速パスを使用することもできます。

このコマンドでリストされたフラグおよび引数の中には、特定のプリンター・タイプに対して無効なものがあります。オプションを使うと問題が発生する場合は、SMIT を使用して **qprt** コマンドのステートメントを実行前に調べることができます。オペレーティング・システムおよびデバイスのマネージの System Management Interface Tool (SMIT) のセクションを参照してください。また、使用中のプリンターがサポートするオプションについては、そのプリンターの解説書を参照してください。

## フラグ

### **-#{j|hlv}**

特殊な機能性を指定します。 *Value* 変数に有効な値は以下のとおりです。

- j** 指定した印刷ジョブのジョブ番号を表示します。
- h** 印刷ジョブをキューに入れますが、HELD 状態で保持します。
- v** 指定したプリンター・バックエンド・フラグの値の妥当性を検査します。妥当性検査プロセスの一部として、このコマンドはフラグ値の違法性検査、タイプ検査、範囲検査、リスト検査、他のタイプの妥当性検査を行います。一般に、バックエンド・フラグ値の妥当性検査を行うと、印刷ジョブが処理される段階になってからではなく、印刷ジョブを実行要求した時点で違法なフラグが識別されるので便利です。

### **-= OutputBin**

印刷ジョブに出力ピンの宛先を指定します。このフラグを指定しないと、プリンター・ドライバーからのデフォルト値が使用されます。

*OutputBin* に使用可能な値は次のとおりです。

- 0** トップ・プリンター・ピン
- 1 - 49** 大容量出力 (HCO) ピン 1 から 49
- >49** プリンター特定出力ピン

注: 有効な出力ピンは、プリンターに依存しています。

**-a PreviewOption**

実際にファイルを印刷せずに印刷ジョブのパラメーター値を事前に調べます。  
*PreviewOption* 変数には、**0** または **1** を指定できます。**0** を指定すると、**qprt** コマンド・プレビューは通常の印刷処理を表示します。**1** を指定すると、コマンドはフラグ値と、入力データ・タイプをプリンターが期待するデータ・タイプに変換するためのフィルター・パイプラインのリストを戻します。これらのフラグ値は、構成データベースから取り出されるデフォルトのコマンド行フラグの値であり、コマンド行でフラグ・パラメーターを指定するとオーバーライドされません。

**-d** に対して指定した (または、デフォルトの) *InputDataType* 変数に有効なフラグだけが表示されます。実際の印刷ではなく印刷ジョブのスプーリングにのみ関連するフラグ値は表示されません。スプーリング・フラグのデフォルト値はそのフラグの説明とともに表示されます。フラグ値がプリンターでサポートされるかどうかは検査されません。

フィルターのパイプラインは、プリンターに送られる前に印刷ファイルからのデータを処理するフィルター・コマンド (およびフィルター・コマンドに渡されるフラグ値) を示します。フィルター・コマンドごとに記述を再検討して、実行されるフィルター操作のタイプを判定できます。

**-A Level**

診断出力のレベルを指定します。診断出力は、フィルターのパイプラインが印刷ファイル、ヘッダー・ページ、またはトレーラー・ページを処理するときにかかるエラーを診断するのに有効です。診断出力は印刷ジョブを実行要求したユーザーにメールされます。以下のいずれかのレベルを指定できます。

- 0** 生成された標準エラー出力を破棄します。
- 1** 標準エラー出力を生成したフラグ値、標準エラー出力、およびパイプライン全体を戻します。
- 2** エラーが検出されるかどうかに関係なく、フラグ値、標準出力エラー (もしあれば)、およびパイプライン全体を戻します。エラーが検出されると、印刷ジョブが終了します。
- 3** 値が **2** の場合と似ていますが、ファイルは印刷されません。

望ましい値は **1** です。パイプライン内のフィルターが標準エラーへの出力を生成する場合は、エラーが検出されなくても (例えば、状況情報が出力されなくても) **0** が有効です。問題が発生したときに標準エラー出力が生成されない場合でも、問題を診断するには **2** か **3** が有効です。

**-b BottomMargin**

下部マージンを指定します。下部マージンとは、各ページの下部に残されるブランク行の数です。



**-B Value**

バースト・ページを印刷します。 *Value* 変数は、2 文字の文字列です。 1 番目の文字はヘッダー・ページに適用されます。 2 番目の文字はトレーラー・ページに適用されます。以下の値が有効です。

- a** 各印刷ジョブ内のファイルごとに必ず (ヘッダーまたはトレーラー) ページを印刷します。
- n** (ヘッダーまたはトレーラー) ページを決して印刷しません。
- g** 各印刷ジョブ (ファイル・グループ) ごとに (ヘッダーまたはトレーラー) ページを印刷します。

例えば、**-B ga** フラグを指定すると、各印刷ジョブの開始時にヘッダー・ページが印刷され、各印刷ジョブの各ファイルの後にトレーラー・ページが印刷されます。

**注:** リモート印刷環境では、デフォルトは、サーバー上のリモート・キューにより求められます。

**-c**

各印刷ファイルをコピーし、コピーから印刷します。このフラグは、**qpri** コマンドが発行された後、ただし印刷ジョブが完了する前に、印刷ファイルまたは複数の印刷ファイルを変更したい場合に指定されます。

このフラグを指定しなかったときに、印刷ジョブが実行要求されたのと同じノード上で印刷される場合、印刷ファイルのコピーは作成されません。 *File* パラメーターで指定したファイルから直接印刷されます。

**-C**

ユーザーがログインしていても、印刷ジョブで生成されたメッセージをそのユーザーにメールします。デフォルトでは、**qpri** コマンドはメッセージをコンソール上に表示します。

**-C** フラグは、ローカル印刷ジョブのみに適用されます。リモート・プリンターに送信されたジョブが完了したときに通知が必要な場合、**-n** フラグを使用してメール・メッセージを受信します。

**注:** どのようなやり方をしても、**qdaemon** およびプリンター・バックエンドからリダイレクトできないメッセージがあります。これらのメッセージは **/dev/console** ファイルに直接送られます。

**-d InputDataType**

印刷する 1 つまたは複数のファイルの入力データ・タイプを識別します。印刷ファイルは、入力データ・タイプとプリンターが期待するデータ・タイプに基づいて、プリンターに送られる前にフィルターを通過 (パススルー) します (必要な場合)。以下のいずれかの入力データ・タイプを指定できます。

- a** 拡張 ASCII
- c** PCL
- d** Diablo 630
- g** Hewlett-Packard GL
- p** パススルー (変更されずにプリンターに送られます)
- s** PostScript

選択したプリンターが指定の入力データ・タイプをサポートしない場合、およびフィルターが使用できないために印刷ファイルのデータ・タイプをプリンターがサポートするデータ・タイプに変換できない場合は、印刷ジョブが終了し、エラー・メッセージが表示されます。

<b>-D "User"</b>	<i>User</i> へ送ることを表すラベルを出力に付けます。通常、出力には <b>qprt</b> コマンド要求を発行している人のユーザー名へ送ることを表すラベルが付けられます。 <i>User</i> の値は、通常のユーザー ID と同じ要求を満たす単一のワードでなければなりません。
<b>-e EmphasizedOpt</b>	強調印刷を以下のいずれかに設定します。 + 強調印刷を使用します。 ! 強調印刷を使用しません。
<b>-E DblHigh</b>	縦倍角印刷を以下のいずれかに設定します。 + 縦倍角印刷を使用します。 ! 縦倍角印刷を使用しません。
<b>-f Filter</b>	印刷ファイルをプリンターに送る前に通過 (パススルー) させるフィルターを識別します。この ID は、 <b>lpr</b> コマンドで使用できるフィルター・フラグに類似しています。使用可能なフィルター ID は、 <b>pr</b> フィルターを起動する <b>p</b> 、 <b>troff</b> コマンドからの出力を処理する <b>n</b> 、および制御文字を印刷できるようにする <b>l</b> です。
<b>-F Name</b>	印刷用に使用される文字イメージが入っている X フォント・ファイルのリストを指定します。リスト内のエントリはコンマで区切らなければなりません。Name パラメーターの値として、絶対パス名、フォント別名、または XLFD 名を使用できます。-F Name フラグは、MBCS プリンター・キューにのみ有効です。
<b>-g Begin</b>	印刷開始ページ番号を指定します。このフラグは、印刷ファイルがフォーマットされている場合 (例えば、-d a フラグを使用する) にしか認識されません。パススルー (-d p フラグ)、PostScript (-d s フラグ)、およびフォーマット済みのその他のタイプのデータについては、このフラグは認識されません。
<b>-G Coord</b>	用紙の端まで印刷できないレーザー・プリンター上でページを印刷する方法を示します。Coordinate 変数には以下のいずれかの値を使用します。 + ページ全体の座標システム。 ! 印刷ページの座標システム。
<b>-h "Header"</b>	-f p フラグも指定されている場合に、pr コマンドで使用するためのヘッダー・テキストを指定します。このフラグを指定しなければ、pr コマンドは印刷ファイル名をヘッダーとして使用します。  このフラグは、-c フラグも指定されている場合に有効です。-c フラグでは、pr コマンドでデフォルト・ヘッダーとして使用される印刷ファイル名は、qprt コマンドで指定されたファイル名に代わって、スプーラーで生成される一時ファイル名になります。
<b>-H "HostName"</b>	ヘッダー・ページ上のホスト名を設定します。
<b>-i Indent</b>	指定したスペース数だけ各行を字下げさせます。-w フラグで指定されたページ幅には、Indent 変数が含まれなければなりません。
<b>-l FontID</b>	(i の大文字) フォント ID を指定します。フォント ID を指定すると、ピッチ (-p フラグ) および書体 (-s フラグ) がオーバーライドされます。-lFontID コマンドは、単一バイト・コード・セットの印刷キューにのみ有効です。
<b>-l FontPath</b>	(i の大文字) は、フォント別名あるいは XLFD 名が付けられたフォント・ファイルを指定する場合に -F フラグに必要なフォント・パスのリストをコンマで区切って指定します。FontPath フラグは、MBCS プリンター・キューにのみ有効です。
<b>-j Init</b>	各ファイルを印刷する前に、プリンターを初期化します。以下のいずれかの値を指定できます。 0 初期化されません。 1 完全に初期化されます。 2 エミュレーター選択のみ。

- J Restore** 印刷ジョブの終了時にプリンターを復元します。以下のいずれかの値を指定できます。
- + 印刷ジョブの終了時に復元します。
  - ! 印刷ジョブの終了時に復元しません。
- k Color** 印刷時の色を指定します。一般的な値は黒、赤、青、緑などです。どのような色がサポートされているか、および特定の色に割り当てられているリボン位置については、プリンターの解説書を参照してください。
- K Condense** 縮刷機能を以下のいずれかに設定します。
- + 縮刷機能を使用します。
  - ! 縮刷機能を使用しません。
- l Length** (L の小文字) ページ長を設定します。 *Length* 変数が 0 であれば、ページ長は無視され、出力は単一の連続ページとして扱われます。ページ長は上下のマージンを含み、用紙の印刷可能長を示します。
- L LineWrap** 1 行がページ幅より長い場合の行のラップを、以下のいずれかの値で設定します。
- + 長い行は次行にラップします。
  - ! 長い行は右マージンで切り捨てます。
- m "Message"** 印刷ジョブがプリンターに割り当てられ、印刷開始準備が整ったときに、コンソールに表示されます。コンソールでメッセージが確認されるまで、印刷ジョブは処理されません。
- M MessageFile** メッセージ・テキストが入っているファイルを識別します。このテキストは、印刷ジョブがプリンターに割り当てられ、印刷開始準備が整ったときに、コンソールに表示されます。コンソールでメッセージが確認されるまで、印刷ジョブは処理されません。
- n** 印刷ジョブの完了時にユーザーに通知します。 **-D "User"** フラグも指定されている場合には、指定したユーザーにも通知されます。 デフォルトでは、印刷ジョブが完了してもユーザーには通知されません。
- N NumberCopies** 印刷する部数を指定します。このフラグを指定しなければ、一部のみが印刷されます。
- O PaperHand** 入力用紙処理のタイプを以下のいずれかの値に設定します。
- 1 手動 (一度に 1 枚ずつ挿入します)
  - 2 連続用紙
  - 3 紙送り
- p Pitch** 1 インチ当たりの文字数を設定します。 *Pitch* の一般的な値は 10 と 12 です。印刷される文字の実際のピッチも、 **-K** (圧縮) フラグと **-W** (横倍角) フラグの値によって影響されます。
- PostScript プリンターで ASCII ファイルを印刷する場合、このフラグは文字のポイント・サイズを決定します。 1 以上の正の数値を指定できます。

- P** *Queue[:QueueDevice]* 印刷キュー名とオプションのキュー・デバイス名を指定します。このフラグが指定されない場合、以下の状況が発生します。
- **LPDEST** 環境変数が設定されている場合は、**qprt** コマンドは **LPDEST** 変数で指定されたキュー名を使用します。この環境変数を設定すれば、**PRINTER** 変数も設定した場合でも、この値が必ず使用されます。
  - **PRINTER** 変数が設定されており、**LPDEST** 変数が設定されていなければ、**qprt** コマンドは **PRINTER** 環境変数で指定されたキュー名を使用します。コマンド行に宛先オプションが指定された場合には、**LPDEST** 環境変数と **PRINTER** 環境変数が両方ともオーバーライドされます。
  - **LPDEST** 変数も **PRINTER** 変数も設定されていない場合、**qprt** コマンドはシステム・デフォルト・キュー名を使用します。(システム・デフォルト・キュー名は、**/etc/qconfig** ファイルで定義された最初のキュー名です。) *QueueDevice* 変数を指定しなければ、キュー用に構成された最初の使用可能プリンターが使用されます。  
 注: 複数のプリンターが同じ印刷キューに構成されており、そのうちの 1 つ以上がユーザーのファイルを印刷するのに適切でない場合には、*QueueDevice* 変数を使用する必要があります。そうでなければ、スーパーが最初に使用可能なプリンターを割り当てます。
- q** *Quality* 印刷品質を以下のいずれかに設定します。
- 0** 高速フォント
  - 1** ドラフト品質
  - 2** レターに近い品質
  - 3** 拡張品質
  - 300** 1 インチ当たり 300 ドット (dpi)
  - 600** 600 dpi
- Q** *Value* 用紙サイズを設定します。用紙サイズのための *Value* は、プリンターに依存しています。一般的な値としては、レターサイズの用紙を表す **1**、リーガル・サイズの用紙を表す **2** などがあります。特定の用紙サイズに割り当てられている値については、該当するプリンターの解説書を参照してください。
- r** 印刷ジョブが完了してから印刷ファイルを除去します。このフラグを指定しなければ、印刷ファイルは除去されません。
- R** *Priority* 印刷ジョブの優先順位を設定します。 *Priority* 変数の値が大きいほど、印刷ジョブの優先順位が上位になります。デフォルトの優先順位値は **15** です。優先順位の最大値は、ほとんどのユーザーについては **20** ですが、 root ユーザー特権を持つユーザーおよびシステム・グループ (グループ 0) については **30** です。  
 注: このフラグは、リモート印刷ジョブを要求する場合には使用できません。
- s** *NameType* *NameType* 変数を使用してタイプ・スタイルを指定します。タイプ・スタイル例としては、**courier** と **prestige** があります。タイプ・スタイルの選択肢は、プリンターのタイプに応じて異なります。
- S** *Speed* 高速印刷を以下のいずれかに設定します。
- +** 高速印刷を使用します。
  - !** 高速印刷を使用しません。
- t** *TopMargin* 上部マージンを指定します。上部マージンとは、各ページの上部に残されるブランク行の数です。

<b>-T "Title"</b>	<i>Text</i> 変数を使用して印刷ジョブのタイトルを指定します。このフラグを指定しなければ、 <b>qprt</b> コマンド行の最初のファイル名が印刷ジョブのタイトルとして使用されます。印刷ジョブのタイトルは、ヘッダー・ページ上、およびキューの状況に関する照会に対する応答に表示されます。
<b>-u PaperSrc</b>	用紙ソースを以下のいずれかに設定します。 <b>1</b> 基本 <b>2</b> 代替 <b>3</b> 封筒
<b>-U Directional</b>	単一方向印刷を以下のいずれかに設定します。 <b>+</b> 単一方向印刷を使用します。 <b>!</b> 単一方向印刷を使用しません。
<b>-v LinesPerIn</b>	行密度を 1 インチ当たりの行数に設定します。 <i>LinesPerIn</i> 変数の一般的な値は <b>6</b> と <b>8</b> です。
<b>-V Vertical</b>	垂直印刷を以下のいずれかに設定します。 <b>+</b> 垂直印刷を使用します。 <b>!</b> 垂直印刷を使用しません。
<b>-w PageWidth</b>	ページ幅を文字数単位で設定します。ページ幅には、 <b>-i</b> フラグで指定されたインデント・スペースの数が含まれなければなりません。
<b>-W DblWide</b>	倍角印刷を以下のいずれかに設定します。 <b>+</b> 倍角印刷を使用します。 <b>!</b> 倍角印刷を使用しません。
<b>-x LineFeed</b>	自動改行または自動復帰を指定します。 <b>0</b> 改行、垂直タブ、復帰を変更しません。 <b>1</b> 復帰ごとに 1 行だけ改行を追加します。 <b>2</b> 改行ごとおよび垂直タブごとに 1 回だけ復帰を追加します。
<b>-X CodePage</b>	コード・ページ名を与えます。 <i>CodePage</i> 変数の有効な値は、ISO8859-1 から ISO8859-9、IBM-943、IBM-eucJP、IBM-eucKR、IBM-eucTW、および UTF-8 です。ユーザーのロケール定義で指定されたコード・ページがデフォルトです。
<b>-y DblStrike</b>	二重印刷を以下のいずれかに設定します。 <b>+</b> 二重印刷を使用します。 <b>!</b> 二重印刷を使用しません。
<b>-Y Duplex</b>	両面印刷を指定します。両面印刷は、用紙の裏表を印刷に使用します。以下のいずれかの値を設定できます。 <b>0</b> 片面印刷 <b>1</b> 両面印刷、長辺がとじ代 <b>2</b> 両面印刷、短辺がとじ代
<b>-z Rotate</b>	ページ・プリンターの出力を、 <i>Value</i> 変数で指定された 1/4 回転の倍数だけ右回りに回転させます。長さ ( <b>-l</b> ) 値と幅 ( <b>-w</b> ) 値は、これに応じて自動的に調整されます。 <b>0</b> 縦長 <b>1</b> 右向き横長 <b>2</b> 下向き縦長 <b>3</b> 左向き横長

## **-Z FormFeed**

各印刷ファイルの後に改ページをプリンターに送ります。以下のいずれかを指定できます。

**+** 改ページ・コマンドを送ります。

**!** 用紙送りコマンドをプリンターに送りません。このオプションを使うと、この印刷ジョブで生成される最後の出力ページから、次の印刷ジョブが始まることがあるので、注意して使ってください。また、連続用紙に印刷するプリンターは、以降のページで用紙の上部を判別できないことがあります。

## 例

1. デフォルト値を使用してデフォルト印刷キュー用に構成した最初の使用可能なプリンター上で印刷するために `myfile` ファイルをキューに入れるには、以下のように入力します。

```
qprt myfile
```

2. ファイルを特定のキューに入れて、デフォルト以外のフラグ値を使用せずに印刷し、印刷ジョブの実行要求時にフラグ値の妥当性を検査するには、以下のように入力します。

```
qprt -f p -e + -P fastest -r -n -C -#v somefile
```

このコマンド行により、`somefile` ファイルが `pr` コマンド (`-f p` フラグ) を通じて渡され、キュー **fastest** (`-P fastest` フラグ) 用に構成された最初の使用可能プリンターで、強調モード (`-e +` フラグ) を使用して印刷されます。`-#v` フラグにより、印刷ジョブがバックエンドに渡される前に、このコマンドに関連するすべてのフラグが有効かどうかを確認されます。ファイルは印刷後に除去され (`-r` フラグ)、印刷ジョブを実行要求したユーザーには印刷ジョブの完了がメール (`-C` フラグ) によって通知されます (`-n` フラグ)。

3. リーガル・サイズの用紙に `myfile` を印刷するには、次のように入力します。

```
qprt -Q2 myfile
```

4. `myfile` ファイルをエンキューしてジョブ番号を戻すには、次のように入力します。

```
qprt -#j myfile
```

5. `MyFile` をキューに入れて保持するには、次のように入力します。

```
qprt -#h MyFile
```

## ファイル

`/etc/qconfig`

キューとキュー・デバイス構成ファイルが入っています。

`/usr/bin/qprt`

`qprt` コマンドが入っています。

## 関連情報

`enq` コマンド、`lp` コマンド、`lpr` コマンド、`pr` コマンド、`qadm` コマンド、`qcan` コマンド、`qchk` コマンド、`qpri` コマンド、`smit` コマンド、`troff` コマンド。

`/etc/qconfig` ファイル。

プリンターおよび印刷の印刷管理のセクション、印刷スプーラーのセクション、プリンター固有の情報 のセクション、仮想プリンターの定義および属性のセクション、初期プリンターの構成のセクション、印刷ジョブを開始するコマンド (`qprt` コマンド) のセクション。



ネットワークおよびコミュニケーションのマネージのリモート・システムからのファイルの印刷のセクション。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

---

## qstatus コマンド

### 目的

印刷スプーリング・システムのプリンター状況を提供します。

### 構文

```
qstatus [ -# JobNumber ] [ -A ] [ -L | -W ] [ -P Printer ] [ -e ] [ -q ] [ -u UserName ] [ -w DelaySeconds ]
```

### 説明

**qstatus** コマンドは、印刷スプーリング・システムの実際の状況機能を実行します。このコマンドは、コマンド行から入力されることはなく、**enq** コマンドによってコールされます。**qstatus** コマンドは、指定されたジョブ、プリンター、キュー、またはユーザーに関する状況情報を生成します。

**qstatus** コマンドによって生成される表示には、リモート・キュー用のエントリーが 2 つ含まれています。第 1 のエントリーには、クライアントのローカル・キューとローカル・デバイス名、その状況情報が含まれます。第 2 のエントリーは第 1 のエントリーのすぐ後にあり、クライアントのローカル・キュー名(再度)の後にリモート・キュー名が表示されます。リモート・キューに対して実行依頼されたジョブは最初にローカル側に表示され、ジョブがリモート・マシン上で処理されるためにリモート・デバイスに移動されます。

状況コマンドはリモート・マシンと通信するので、リモート・マシンからの応答を待機する間に状況表示が停止しているように見えることが時々あります。2 つのコンピューターの間で接続が確立できなければ、コマンドは最終的にタイムアウトになります。

### フラグ

フラグはすべてオプションです。フラグを指定しなければ、**qstatus** コマンドは以下の状況を戻します。

- **LPDEST** 環境変数を設定した場合は、**LPDEST** 変数で指定されたプリンター。この環境変数を設定すれば、**PRINTER** 変数も設定した場合でも、この値が必ず使用されます。
- **PRINTER** 変数を設定し **LPDEST** 変数を設定しなかった場合は、**PRINTER** 環境変数が指定されたプリンター。
- **LPDEST** 変数も **PRINTER** 変数も設定しなかった場合は、デフォルトのプリンター。

注: 宛先コマンド行オプションがあった場合には、**LPDEST** 環境変数と **PRINTER** 環境変数が両方とも無効になります。



<b>-#JobNumber</b>	<i>JobNumber</i> 変数で指定されたジョブについて、現在の状況情報を表示します。通常は、キューに入れられているすべてのジョブの状況が表示されます。 1. デフォルトの宛先プリンターをオーバーライドするには、 <b>-P Queue</b> を指定します。 2. ジョブ 1、2、および 3 がプリンター・キューに入っている場合、ジョブ 1 の実行時にジョブ 3 の状況を表示するように指定しても、状況情報はジョブ 3 だけでなく、ジョブ 1 とジョブ 3 を表示します。 3. 存在しないジョブ番号を指定すると、システムはエラー・メッセージの代わりに、キュー上の現行ジョブ番号を表示します。
<b>-A</b>	<b>/etc/qconfig</b> ファイル内で定義されたすべてのキューに関する状況情報が表示されます。
<b>-e</b>	<b>qdaemon</b> コマンドが制御していないキューからの状況情報を除外します。このようなキューの状況は、別のフォーマットで表示されることがあります。 <b>-e</b> フラグは、他のフラグと任意に組み合わせて使用できます。
<b>-L</b>	コマンド行で要求された状況情報について詳細に表示します。このフラグは <b>-W</b> フラグとともに使用することはできません。 <b>-L</b> フラグと <b>-W</b> フラグを同時に使用する場合、最初に指定されたフラグが優先されます。
<b>-P Printer</b>	<i>Printer</i> 変数で指定されたプリンターに関する現在の状況情報を表示します。通常は、デフォルト・プリンターか、 <b>LPDEST</b> 環境変数または <b>PRINTER</b> 環境変数のいずれかの値が使用されます。常に <b>LPDEST</b> 変数が <b>PRINTER</b> 変数に優先します。
<b>-q</b>	デフォルト・キューの現在の状況を表示します。デフォルト・キューは、 <b>LPDEST</b> 変数で指定するか、または <b>LPDEST</b> 値がない場合は <b>PRINTER</b> 環境変数で指定します。どちらの変数もない場合には、 <b>qstatus</b> コマンドは、 <b>/etc/qconfig</b> ファイルにリストされている最初のキューを使用します。
<b>-u UserName</b>	<i>UserName</i> 変数で指定されたユーザーが実行依頼したすべてのジョブの現在の状況情報を表示します。通常は、キューに入れられているすべてのジョブの状況が表示されます。
<b>-W</b>	これを指定すると、さらに長いキュー名、デバイス名、およびジョブ番号を持つ広幅バージョンの状況情報が表示されます。より長いジョブ番号情報は、AIX バージョン 4.3.2 以降で提供されます。このフラグは <b>-L</b> フラグとともに使用することはできません。 <b>-L</b> フラグと <b>-W</b> フラグを同時に使用する場合、最初に指定されたフラグが優先されます。
<b>-w DelaySeconds</b>	要求されたキュー情報を、 <i>DelaySeconds</i> 変数で指定された間隔で表示します。キューが空であれば表示が終了します。

## 例

1. デフォルトの印刷キューを表示するには、次のように入力します。

```
qstatus -q
```

2. 空になるまで 5 秒ごとに画面を更新しながら、すべてのキューの長時間状況を表示するには、次のように入力します。

```
qstatus -A -L -w 5
```

3. プリンター lp0 の状況を表示するには、次のように入力します。

```
qstatus -P lp0
```

4. ジョブ番号 123 の状況を表示するには、次のように入力します。

```
qstatus -# 123 -P lp0
```

5. AIX 4.2.1 またはそれ以降のバージョンのすべてのキューの状況を広幅フォーマットで表示するには、次のように入力します。

```
qstatus -A -W
```

## ファイル

<code>/var/spool/lpd/qdir/*</code>	ジョブ記述ファイルが入っています。
<code>/etc/qconfig</code>	構成ファイルが入っています。
<code>/etc/qconfig.bin</code>	<code>/etc/qconfig</code> ファイルの要約されたバイナリー・バージョンが入っています。
<code>/usr/lib/lpd/rembak</code>	リモート・バックエンドが入っています。
<code>/usr/lib/lpd/qstatus</code>	コマンド・ファイルが入っています。
<code>/var/spool/lpd/stat/*</code>	<code>qstatus</code> コマンド用の状況ファイルが入っています。

## 関連情報

`enq` コマンド、`lpd` コマンド、`lpr` コマンド、`qdaemon` コマンド、`qchk` コマンド、`rembak` コマンド。

`/etc/qconfig` ファイル。

プリンターおよび印刷 の 印刷スプーラー のセクション。

プリンターおよび印刷 の リモート印刷 のセクション。

---

## quiz コマンド

### 目的

ユーザーの知識をテストします。

### 構文

```
quiz { -i File | -t | Category1 Category2 }
```

### 説明

`quiz` コマンドは、さまざまな選択可能対象について連想知識テストを行います。`Category1` から選択されたエントリーについて質問し、`Category2` からの解答を期待します。カテゴリーを指定しないと、`quiz` コマンドは使用可能なカテゴリーをリストし、指示を出してから、シェル・プロンプトに戻ります。

Enter キーを押すたびに、ゲームは正解を表示します。問題がなくなるか、割り込み (Ctrl-C) キー・シーケンスまたはファイル終わり (Ctrl-D) キー・シーケンスを押すと、ゲームは得点を報告して終了します。

## フラグ

- i File** 標準索引ファイルの代わりに、指定された *File* を使用します。
- 注: 以下の構文説明では、大括弧は一般にエントリーがオプションであることを示します。ただし、太字体の大括弧または中括弧は、構文のリテラル部分として入力しなければなりません。エントリーの垂直リストは、そこから 1 つ選択しなければならないことを示します。 *File* 内の行には、以下の構文を使用しなければなりません。
- ```
line      = category [:category] . . .
category = alternate [ |alternate ] . . .
alternate = [primary]
primary   = character
           [category]
           option
option    = {category}
```
- 索引ファイル内で、各行の最初のカテゴリは情報ファイル名を指定しなければなりません。情報ファイルには、クイズ問題を収めたファイルの名前が入っています。それ以外のカテゴリは、情報ファイルの各行に保管されているデータの順序と内容を指定します。情報ファイル内のクイズ・データには、同じ構文を使用します。
- \ (バックスラッシュ) はエスケープ文字です。これを使用すると、構文上重要な文字を引用符で囲んだり、行に改行文字 (\n) を挿入できます。問題またはその解答がブランクであれば、**quiz** コマンドは問題を出しません。構造 **alab** は情報ファイル内では機能しません。 **a{b}** を使用してください。
- t** チュートリアルを提供します。誤答された問題を繰り返し、次々に新しい問題を紹介します。

## 例

1. ラテン語の英語訳に関するクイズを開始するには、次行を入力します。

```
/usr/games/quiz latin english
```

ゲームは、ラテン語の単語を表示して、ユーザーが英語の訳を入力するまで待機します。

2. 英語のラテン語訳に関するクイズを開始するには、次行を入力します。

```
/usr/games/quiz english latin
```

3. ラテン語の英語訳に関するクイズを設定するには、索引ファイルに次行を追加します。

```
/usr/games/lib/quiz/latin:latin:english
```

この行は、**/usr/games/lib/quiz/latin** ファイルにラテン語と英語のカテゴリに関する情報が入っていることを指定します。

新しいカテゴリは、標準索引ファイル **/usr/games/lib/quiz/index** またはユーザー独自の索引ファイルに追加できます。ユーザー独自の索引ファイルを作成する場合は、**-iFile** フラグを付けて **quiz** コマンドを実行し、クイズのトピック・リストを入力します。

4. 以下の例はサンプルの情報ファイルです。

```
cor:heart
sacerdos:priest{ess}
quando:when|since|because
optat:{{s}he |it |[desires|wishes]}\
desire|wish
alb[us|a|um]:white
```

この情報ファイルにはラテン語と英語の単語が入っています。各ラテン語の単語と英語訳は : (コロン) で区切られます。 { } (中括弧) で囲まれたエントリーはオプションです。 | (縦線) で区切られたエントリーは、どちらを入力しても正解です。縦線で区切られたエントリーをグループ化するには、 [ ] (大括弧) を使用します。

1 行目は、ラテン語の単語 `cor` への応答として解答 `heart` だけが受け入れられます。2 行目は、`sacerdos` への応答として `priest` または `priestess` を受け入れます。3 行目は、`quando` に対して `when`、`since`、または `because` を受け入れます。

4 行目の末尾に付いている \ (バックスラッシュ) は、このエントリーが次行まで継続することを示します。つまり、4 行目と 5 行目は、2 行で 1 つのエントリーとなっています。このエントリーには、`optat` への応答として以下のいずれかを受け入れます。

```
she desires it desires desire
she wishes it wishes wish
he desires desires
he wishes wishes
```

ラテン語の英語訳クイズを開始すると、サンプル情報ファイルの最終行が **quiz** コマンドに対して、ラテン語の単語 `albus` の意味をユーザーに尋ねるように命令します。英語のラテン語訳クイズを開始すると、**quiz** コマンドは `white` を表示して、解答として `albus`、`alba`、または `album` を受け入れます。

問題エントリーに文字 { (左中括弧)、} (右中括弧)、[ (左大括弧)、] (右大括弧)、または | (縦線) があると、**quiz** コマンドは各 | グループの最初の代替を示し、各オプション・グループを表示します。したがって、このサンプル内の第 4 の定義の場合、英語のラテン語訳の問題は `she desires` となります。

## ファイル

|                                        |                         |
|----------------------------------------|-------------------------|
| <code>/usr/games/lib/quiz/index</code> | クイズ・カテゴリー用のデフォルト索引ファイル。 |
| <code>/usr/games/lib/quiz/*</code>     | 特定ファイルの内容を指定する時に使用します。  |
| <code>/usr/games</code>                | システムのゲームの位置。            |

## 関連情報

**arithmetic** コマンド、**back** コマンド、**bj** コマンド、**craps** コマンド、**fish** コマンド、**fortune** コマンド、**hangman** コマンド、**moo** コマンド、**number** コマンド、**ttt** コマンド、**turnoff** コマンド、**turnon** コマンド、**wump** コマンド。

---

## quot コマンド

### 目的

ファイルシステムの所有権を要約します。

### 構文

```
quot [ -c ] [ -f ] [ -h ] [ -n ] [ -v ] [ FileSystem ... ]
```

```
quot -a [ -c ] [ -f ] [ -h ] [ -n ] [ -v ]
```

## 説明

**quot** コマンドは、指定したファイルシステム (*FileSystem*) に各ユーザーが現在所有している 512 バイトのブロックの数を表示することにより、ファイルシステムの所有権を要約します。ファイルシステムを指定しないと、**quot** コマンドは、**/etc/filesystems** ファイル内のファイルシステムごとに同じ情報を表示します。

## フラグ

- a** マウントされたすべてのシステムのレポートを生成します。
- c** 3 列のレポートを表示します。1 列目にはファイル・サイズが 512 バイトのブロック単位で指定されます。2 列目にはそのサイズのファイル数が指定されます。最後に、3 列目にはそのサイズ以下のすべてのファイル内の 512 ブロックの累積合計が指定されます。  
注: 500 ブロック以上のファイルは、ブロック・サイズ 499 の下にグループ化されます。ただし、それらの正確なブロック数はブロックの累積合計に送られます。
- f** ブロックの総数、ファイルの総数、およびこれらの合計に対応するユーザー名を表示します。
- h** ファイルによって使用されるブロック数を予測します。この見積もりはファイル・サイズに基づいており、ホール付きファイルの使用時には実際のブロック使用量よりも大きい値を戻す場合もあります。
- n** 以下のパイプラインを実行することによって、すべてのファイルとそのオーナーのリストを作成します。  
`ncheck filesystem | sort +0n | quot -n filesystem`
- v** 直近の 30 日、60 日、および 90 日にアクセスされなかったブロックの数を、3 列の出力に表示します。

## セキュリティ

アクセス制御: このコマンドは `bin` ユーザーと `bin` グループによって所有されています。

## 例

1. **/usr** ファイルシステム内の各ユーザーによって所有されているファイル数とバイト数を表示するには、以下のように入力します。

```
quot -f /usr
```

システムは以下の情報を表示します。

```
/usr:
63056 3217 bin
20832 390  root
1184  42   uucp
 56   5    adm
 8    1   guest
 8    1   sys
```

2. ファイル・サイズ統計、各サイズのファイル数、および累積合計を表示するには、次のように入力します。

```
quot -c /usr
```

システムは以下の情報を表示します。

```
/usr:
8      103   824
16     2    856
499    0    856
```

3. マウントされたすべてのファイルシステムのレポートを生成するには、次のように入力します。

```
quot -a
```

4. **/var** ファイルシステムのレポートを生成するには、次のように入力します。

```
#quot -v /var
/var:
45695 root          12852  11878  11774
2569  guest          2567   1280   960
2121  adm             92      91     91
1343  bin             465     233    193
14    uucp            0        0      0
5     daemon         0        0      0
1     invscout      1        1      1
1     nuucp         1        1      1
1     sys           0        0      0
```

## ファイル

`/etc/passwd` ユーザー名が入っています。  
`/etc/filesystems` ファイルシステムの名前と位置が入っています。

## 関連情報

`du` コマンド、`ls` コマンド。

---

## quota コマンド

### 目的

ディスク使用状況とクォータを表示します。

### 構文

```
quota [ -u [ User ] ] [ -g [ Group ] ] [ -v | -q ]
```

### 説明

`quota` コマンドは、ディスク使用状況とクォータを表示します。デフォルトの場合、または `-u` フラグを指定した場合は、ユーザー・クォータのみ表示されます。`quota` コマンドは、`/etc/filesystems` ファイル内にリストされているファイルシステムすべてのクォータを報告します。`quota` コマンドがゼロ以外の状況で終了した場合、1 つ以上のファイルシステムがクォータを超えていることとなります。

`root` ユーザーは、`-u` フラグをオプションの `User` パラメーターとともに使用して、他のユーザーの制限値を見ることができます。`root` ユーザー権限を持っていないユーザーは、`-g` フラグをオプションの `Group` パラメーターとともに使用することによって、そのユーザーがメンバーとなっているグループの制限値を見ることができます。

#### 注:

1. JFS ファイルシステムでは、特定のユーザーがクォータを持っているファイルシステム内にファイルを持っていない場合、このコマンドはそのユーザーに対して `quota: none` を表示します。ユーザーの実際のクォータが表示されるのは、ファイルシステム内にそのユーザーのファイルがあるか、または `-v` フラグが指定されたときです。JFS2 の場合、すべてのケースでユーザーの実際のクォータが表示されます。
2. JFS2 システムでは、`root` ユーザーはクォータによって制限を受けることがないため、`root` ユーザーの限界は常にゼロ (無制限) として表示されます。
3. `rpc.rquotad` プロトコルは、NFS のグループ・クォータをサポートしていません。このため、このプロトコルは NFS に対するグループ・クォータ情報を戻しません。

## フラグ

- g ユーザーのグループのクォータを表示します。
- u ユーザーのクォータを表示します。このフラグはデフォルトのオプションです。
- v ストレージが割り当てられていないときのファイルシステム上のクォータを表示します。
- q 使用量がクォータを超えているファイルシステムについての情報のみを示す簡潔なメッセージを表示します。  
注: -q フラグは -v フラグに優先します。

## セキュリティ

アクセス制御: このコマンドは root ユーザーと bin グループによって所有されています。

特権制御: このプログラムは、非特権ユーザーが個人クォータを見ることができるよう、**setuid** するようになっています。

## 例

1. ユーザー keith のクォータを表示するには、次のように入力します。

```
quota
```

システムは以下の情報を表示します。

```
User quotas for user keith (uid 502):
Filesystem blocks quota limit grace Files quota limit grace
/u          20    55    60          20    60    65
```

2. ユーザー davec のクォータを root ユーザーとして表示するには、次のように入力します。

```
quota -u davec
```

システムは以下の情報を表示します。

```
User quotas for user davec (uid 2702):
Filesystem blocks quota limit grace files quota limit grace
/u          48    50    60           7    60    60
```

## ファイル

|                         |                        |
|-------------------------|------------------------|
| <b>quota.user</b>       | ユーザー・クォータを指定します。       |
| <b>quota.group</b>      | グループ・クォータを指定します。       |
| <b>/etc/filesystems</b> | ファイルシステムの名前と位置が入っています。 |

## 関連情報

セキュリティのクォータ・システムのセクション。

**edquota** コマンド、**quotacheck** コマンド、**quotaon** および **quotaoff** コマンド、**repquota** コマンド。

---

## quotacheck コマンド

### 目的

ファイルシステム・クォータの整合性を検査します。



## 構文

```
quotacheck [ -d ] [ -g ] [ -u ] [ -v ] { -a | Filesystem ... }
```

## 説明

**quotacheck** コマンドは、*FileSystem* パラメーターによって指定されたファイルシステムを検査し、現在のディスク使用状況テーブルを作成し、このテーブルの中の情報をファイルシステムのディスク・クォータ・ファイルに記録されている情報と比較します。情報の不一致が検出された場合には、クォータ・ファイルが更新されます。デフォルトでは、ユーザー・クォータとグループ・クォータが両方とも検査されます。

オプションの **-g** フラグは、グループ・クォータのみを検査することを指定します。オプションの **-u** フラグは、ユーザー・クォータのみを検査することを指定します。**-g** フラグと **-u** フラグの両方を指定すると、ユーザー・クォータとグループ・クォータの両方を検査するデフォルト・オプションの動作と同じになります。**-a** フラグは、**/etc/filesystem** ファイル内にある、ディスク・クォータが使用可能なすべてのファイルシステムを検査することを指定します。

ジャーナル・ファイルシステム (JFS) の場合に限って、オプションの **-d** フラグは、**/etc/passwd** または **/etc/group** 内に存在せず、またファイルシステム内に割り振りを持たないすべてのユーザーまたはグループ ID の使用率統計を削除します。これにより、この影響を受けるユーザーまたはグループの統計は、**repquota** コマンドで表示することはできません。

**quotacheck** コマンドは通常、動作後の結果を報告しません。**-v** フラグを指定すると、**quotacheck** コマンドは計算上のディスク・クォータと記録されたディスク・クォータとの間の矛盾を報告します。

JFS の場合は、**quotacheck** コマンドは **/etc/filesystems** ファイルからクォータ・ファイル名を判別します (デフォルトでは、ファイルシステムの root にある、**quota.user** および **quota.group** という名前のファイル)。JFS2 の場合は、これらのファイルの名前と場所は既定されており、変更できません。これらのファイルが存在しなかった場合、**quotacheck** コマンドはこれらのファイルを作成します。

注: アクティブなファイルシステムに対して **quotacheck** コマンドを実行してはなりません。ファイルシステムが現在アクティブのときは、**quotacheck** を実行すると正しくないディスク使用状況情報が出されることがあります。

## フラグ

- a** **/etc/filesystems** にある、ディスク・クォータが使用可能なすべてのファイルシステムを検査します。
- d** 未定義で、割り振りを持たない ID の使用率統計を削除します (JFS のみ)。
- g** グループ・クォータのみを検査します。
- u** ユーザー・クォータのみを検査します。
- v** 計算上のディスク・クォータと記録されたディスク・クォータとの間の相違を報告します。

## セキュリティ

アクセス制御: root ユーザー権限を持つユーザーのみがこのコマンドを実行できます。

## 例

1. /usr ファイルシステムのユーザー・クォータとグループ・クォータを検査するには、次のように入力します。

```
quotacheck /usr
```

2. /usr ファイルシステムのグループ・クォータのみを検査するには、次のように入力します。

```
quotacheck -g /usr
```

## ファイル

|                         |                        |
|-------------------------|------------------------|
| <b>quota.user</b>       | ユーザー・クォータを指定します。       |
| <b>quota.group</b>      | グループ・クォータを指定します。       |
| <b>/etc/filesystems</b> | ファイルシステムの名前と位置が入っています。 |
| <b>/etc/group</b>       | 基本グループ属性が入っています。       |
| <b>/etc/passwd</b>      | ユーザー名が入っています。          |

## 関連情報

**edquota** コマンド、**quota** コマンド、**quotaon** および **quotaoff** コマンド、**repquota** コマンド。

セキュリティのクォータ・システムのセクション。

---

## quotaon または quotaoff コマンド

### 目的

ファイルシステム・クォータをオンおよびオフに切り替えます。

### 構文

```
quotaon [ -g ] [ -u ] [ -v ] { -a | FileSystem ... }
```

```
quotaoff [ -g ] [ -u ] [ -v ] { -a | FileSystem ... }
```

### 説明

**quotaon** コマンドは、*FileSystem* パラメーターで指定した 1 つ以上のファイルシステムのディスク・クォータを使用可能にします。指定されたファイルシステムは、**/etc/filesystems** ファイル内にクォータを定義し、かつマウントしておく必要があります。**quotaon** コマンドは、関連ファイルシステムのルート・ディレクトリー内に **quota.user** および **quota.group** ファイルがないか探し、見付からない場合はエラーを戻します。

注: JFS の場合のみ、デフォルト・クォータ・ファイル名 (**quota.user** および **quota.group**) を **/etc/filesystems** ファイル内でオーバーライドすることができます。クォータ・ファイルは、**/etc/filesystems** ファイル内でフルパスを指定することにより、クォータが使用可能なファイルシステムの外部に存在することが可能です。JFS2 ファイルシステムの場合は、ファイル名をオーバーライドすることはできず、ファイルシステムのルート・ディレクトリーにあることが必要です。

デフォルトでは、ユーザー・クォータとグループ・クォータの両方が使用可能になります。**-u** フラグはユーザー・クォータのみを使用可能にし、**-g** フラグはグループ・クォータのみを使用可能にします。**-g** フラグと **-u** フラグの両方を指定することは、デフォルト (オプションの指定なし) と同じことになります。**-a** フラグは、**/etc/filesystems** ファイルで示されているような、ディスク・クォータを持つファイルシステムのすべてを使用可能にすることを指定します。

**quotaoff** コマンドは、1 つ以上のファイルシステムのディスク・クォータを使用不可にします。デフォルト時は、ユーザー・クォータとグループ・クォータの両方が使用不可になります。**-a** フラグ、**-g** フラグ、および **-u** フラグは、**quotaon** コマンドを使用したときと同じように動作します。**-v** フラグは、

**quotaon** コマンドと **quotaoff** コマンドでそれぞれクォータがオンまたはオフにされるすべてのファイルシステムで、各クォータ・タイプ (ユーザーまたはグループ) ごとに、メッセージを出力します。

**quota.user** および **quota.group** ファイルが、ユーザー **root** およびグループ **system** により所有されていない場合は、エラー (**EPERM**) が戻されます。これらのファイルの所有権の変更は、**quotas** がアクティブである間は、行えません。

## フラグ

- a** **/etc/filesystems** ファイルで示されているような、読み書きされ、しかもディスク・クォータを持っているすべてのファイルシステムを使用可能または使用不可にします。 **-g** フラグと使用すると、**/etc/filesystems** ファイルのグループ・クォータのみが使用可能または使用不可になります。また、**-u** フラグと使用すると、**/etc/filesystems** ファイルのユーザー・クォータのみが使用可能または使用不可になります。
- g** グループ・クォータのみを使用可能または使用不可にすることを指定します。
- u** ユーザー・クォータのみを使用可能または使用不可にすることを指定します。
- v** クォータがオンまたはオフにされるファイルシステムごとにメッセージを出力します。

## セキュリティ

アクセス制御: **root** ユーザーのみがこのコマンドを実行できます。

## 例

1. **/usr** ファイルシステムのユーザー・クォータを使用可能にするには、次のように入力します。  

```
quotaon -u /usr
```
2. **/etc/filesystems** ファイル内の全ファイルシステムのユーザー・クォータとグループ・クォータを使用不可にし、メッセージを出力するには、次のように入力します。  

```
quotaoff -v -a
```

## ファイル

|                         |                        |
|-------------------------|------------------------|
| <b>quota.user</b>       | ユーザー・クォータを指定します。       |
| <b>quota.group</b>      | グループ・クォータを指定します。       |
| <b>/etc/filesystems</b> | ファイルシステムの名前と位置が入っています。 |

## 関連情報

**quota** コマンド、**edquota** コマンド、**repquota** コマンド、**quotacheck** コマンド。

セキュリティのディスク・クォータ・システムの概要のセクションは、ディスク・クォータ・システムについて説明しています。

セキュリティのクォータ・システムのセクション。

---

## raddbm コマンド

### 目的

**RADIUS** ユーザー認証情報のローカル・データベース内の項目を変更します。

## 構文

```
raddbm [ -a Command ] [ -d Database_filename ] [ -e EAP_type ] [ -i Config_filename ] [ -l Load_filename ] [ -n ] [ -p ] [ -t pwd_expire_wks ] [ -u User_ID ] [ -w ]
```

## 説明

**raddbm** コマンドは、ユーザー認証情報のローカル・データベースを作成および変更するために使用します。RADIUS サーバーは、このデータベースをユーザー認証用の情報ソースとして使用するよう構成できます。

このローカル・データベースはファイルに保管されます。ファイル内のデータは、高速検索を可能にするために二分木フォーマットになっています。データベース・ファイル名は RADIUS **/etc/radius/radiusd.conf** 構成ファイルに指定し、デフォルト値は **dbdata.bin** です。このファイル名を変更するには、SMIT を使用して **radiusd.conf** を編集します。

各エントリーには、以下のフィールドがあります。

|                     |                         |
|---------------------|-------------------------|
| USERID              | ユーザーの ID を指定します。        |
| PASSWORD            | ユーザーのパスワードを指定します。       |
| PASSWORD_EXPIRATION | パスワードの満了時間を週数で指定します。    |
| EAP_TYPE            | 認証に使用できる EAP タイプを指定します。 |

データベース・ファイルのパスワードは、単純なパスワード漏えいを防止するためにクリア・テキストでは保管されませんが、パスワードを隠すために使用されるアルゴリズムは暗号的にセキュアであるとはみなされていません。ファイル **dbdata.bin** は、所有者およびグループとして **root: セキュリティー** によって保護されます。

**raddbm** コマンドでは、以下の操作を含むローカル・データベースに対するいくつかの操作がサポートされています。

- ユーザーをデータベースに追加する。

ユーザーを追加するためのコマンド形式は次のとおりです。

```
raddbm -a ADD -u User_ID -e EAP_type -t pwd_expire_wks
```

ユーザーのパスワードは標準入力からプロンプトが出されます。

**-e** および **-t** フラグはオプションです。**-e** フラグに値が入力されない場合、EAP\_TYPE としてデフォルト値の **none** が使用されます。これは、このユーザーに関しては EAP パケットが無視されるということです。**-t** フラグに値が入力されない場合、PASSWORD\_EXPIRATION としてデフォルト値の **0** が使用されます。これは、パスワードの有効期限が一切検査されないということです。**-p** フラグはオプションです。これは、**raddbm** コマンドが常に、新規ユーザーの追加時に新規パスワードの入力を求めるプロンプトを出すためです。

- データベース内のユーザーを変更する。

ローカル・データベース内のユーザーの情報を変更するには、次のように入力します。

```
raddbm -a CHANGE -u User_ID -p -e EAP_type -t pwd_expire_wks
```

**-e**、**-p**、および **-t** フラグはオプションですが、少なくとも 1 つは指定する必要があります。**-p** フラグを使用する場合、**raddbm** コマンドによってパスワードの入力を求めるプロンプトが出されます。

- ユーザーをデータベースから削除する。

データベースからユーザーのエントリを削除するには、次のように入力します。

```
raddbm -a DELETE -u User_ID
```

- データベースのユーザーをリストする。

データベース内のユーザーのエントリをリストするには、次のように入力します。

```
raddbm -a LIST
raddbm -a LIST -u User_ID
raddbm -a LIST -u User_ID -w
```

**-w** および **-u** フラグはオプションです。**-w** フラグを指定すると、ユーザーのエントリのすべてのフィールドが表示されます (ただし、パスワードはセキュリティ上の理由で表示されません)。

**-u** フラグを指定すると、ユーザーの情報がコロンで区切られたフォーマットで表示されます。**-u** フラグを指定しない場合は、データベース内のすべてのエントリが列フォーマットで表示されます。

- 新しいデータベースを作成する。

RADIUS サーバーは、空のデータベースを **/etc/radius/dbdata.bin** に入れて提供します。ユーザーが新しいデータベースを作成する場合、作成時に必ず 1 つ以上のユーザーを追加する必要があります。このコマンドの形式は次のとおりです。

```
raddbm -a ADD -u User_ID -e EAP_type -t pwd_expire_wks -n
```

ユーザーのパスワードは標準入力からプロンプトが出されます。

**-e** および **-t** フラグはオプションです。これらのデフォルトは、それぞれ **EAP\_type=NONE** およびパスワードの有効期限検査なしです。

- ユーザーのリストをデータベースにロードする。

ユーザーのリストは、**-l** フラグを使用するとデータベースに直接ロードできます。データベースにレコードを持つユーザーごとに、次の形式でファイルを 1 つずつ作成する必要があります。

```
"userid" "password"
```

この二重引用符は必要です。

これで、**-l** フラグを使用して次の方法でこのファイルを使用できます。

```
raddbm -l filename
```

ユーザー・パスワードをプレーン・テキスト・フォーマットでファイルに保管するのは、できる限り避けてください。このオプションは、主にテストの目的で提供されています。

## フラグ

|                             |                                                                                                            |
|-----------------------------|------------------------------------------------------------------------------------------------------------|
| <b>?</b>                    | ヘルプ画面を表示します。                                                                                               |
| <b>-a Command</b>           | 実行するアクションを指定します。値は <b>ADD</b> 、 <b>LIST</b> 、 <b>DELETE</b> 、または <b>CHANGE</b> です。                         |
| <b>-d Database_filename</b> | データベース・ファイル名を指定します。 <b>radiusd.conf</b> RADIUS 構成ファイルに指定されたデフォルトのデータベース・ファイルをオーバーライドするときに使用します。            |
| <b>-e EAP_type</b>          | ユーザーが認証用に使用できる EAP タイプを指定します。現在、 <b>MD5-challenge</b> または <b>none</b> のみがサポートされています。デフォルトは <b>none</b> です。 |
| <b>-i Config_filename</b>   | RADIUS 構成ファイル名を指定します。デフォルトの <b>/etc/radius/radiusd.conf</b> 構成ファイルをオーバーライドするときに使用します。                      |
| <b>-l Load_filename</b>     | ロードするユーザー名とパスワードのファイルのファイル名を指定します。                                                                         |

|                                 |                                                                                                                                 |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| <b>-n</b>                       | 新しいデータベース・ファイルを作成します。 <b>ADD</b> コマンド・オプションを指定する場合にのみ有効です。このオプションを使用する場合、データベース内の前の情報がすべて失われます。                                 |
| <b>-p</b>                       | ユーザーのパスワードを変更することを示します。セキュリティ上の理由により、パスワードは、コマンド行からの読み取りではなく、標準入力からプロンプトが出されません。                                                |
| <b>-t</b> <i>pwd_expire_wks</i> | ユーザーのパスワードが有効である週数を指定します。このフラグは、 <b>ADD</b> と <b>CHANGE</b> コマンドを指定する場合にのみ有効です。デフォルトは、パスワードの有効期限がないことを示す 0 です。有効な値は 0 から 52 です。 |
| <b>-u</b> <i>User_ID</i>        | ユーザーの ID を指定します。有効なユーザー ID の長さは 253 文字未満で、英字、数字、および一部の特殊文字を使用できます。ブランクは含まれません。ユーザー ID の重複は認められません。                              |
| <b>-w</b>                       | ユーザー情報の詳細なリストを生成します。                                                                                                            |

## 終了状況

このコマンドには次の終了値があります。

|    |                  |
|----|------------------|
| 0  | コマンドは正常に実行されました。 |
| >0 | エラーが発生しました。      |

## セキュリティ

このコマンドは、root ユーザーまたはセキュリティ・グループのメンバーのみが実行できます。

## 例

1. 新しいローカル RADIUS データベースを作成するには、必ず 1 人以上のユーザーを追加する必要があります。データベースを作成する場合は、次のように入力します。

```
raddbm -a ADD -u user01 -n
```

**注:** **-n** オプションは既存のデータベースを上書きし、前の内容をすべて破棄します。作成されるデータベース・ファイルには、**/etc/radius/radiusd.conf** RADIUS 構成ファイルに指定されたデフォルト名が付けられます。

2. データベースにユーザーを追加するには、次のように入力します。

```
raddbm -a ADD -u user01
```

デフォルト値の EAP\_TYPE = "none" と PASSWORD\_EXPIRATION = "0" が使用されます。

3. データベースからユーザーを削除するには、次のように入力します。

```
raddbm -a DELETE -u user01
```

4. ユーザーのパスワードを変更するには、次のように入力します。

```
raddbm -a CHANGE -u user01 -p
```

このコマンドによって、新しいパスワードの入力を求めるプロンプトが出されます。

5. デフォルト・データベースのすべてのエントリーを含む詳細なリストを表示するには、次のように入力します。

```
raddbm -a LIST -w
```

パスワードは表示されません。



6. 特定のユーザーのデータベース・エントリを表示するには、次のように入力します。

```
raddbm -a LIST -u user01 -w
```

7. ファイルからユーザーのリストを追加するには、最初に 1 行に 1 つずつエントリを含むユーザーとパスワードのファイルを作成します。形式は次のとおりです。

```
"userid" "password"
```

この後、次のように入力します。

```
raddbm -l Load_filename
```

## 制約事項

**raddbm** コマンドを実行する前に、必ず **RADIUS** デーモンを停止してください。このデーモンを停止するには、**stopsrc -s radiusd** コマンドを使用します。データベースを変更した場合は、**startsrc -s radiusd** コマンドを使用してこのデーモンを再始動してください。

## 実装上の固有な条件

このコマンドは **radius.base** ファイルセットに含まれます。

## 位置

**/usr/radius/bin/raddbm**

## 標準入力

セキュリティ上の理由により、ユーザーをデータベースに追加する場合、ユーザーのパスワードはコマンド行からではなく標準入力から読み取られます。

## 標準エラー

**raddbm** コマンドの呼び出しが失敗すると、標準エラーに通知メッセージが書き込まれます。

## ファイル

**/usr/radius/bin/raddbm**

**/etc/radius/raddbm.bin**

**/etc/radius/radiusd.conf**

**raddbm** コマンドの位置。

**radiusd.conf** ファイルに指定されたデフォルトのデータベース・ファイル。

デフォルトのデータベース・ファイル名を含む、RADIUS 構成値を指定します。

## 関連情報

セキュリティのセキュア・システムのインストールおよび構成のセクション。

---

## ranlib コマンド

### 目的

アーカイブ・ライブラリーをランダム・ライブラリーに変換します。



## 構文

**ranlib** [ **-t** ] [ **-X** {**32**|**64**|**32\_64**}] *Archive* ...

## 説明

**ranlib** コマンドは、各 *Archive* ライブラリーをランダム・ライブラリーに変換します。ランダム・ライブラリーとは、記号テーブルが入っているアーカイブ・ライブラリーのことです。

**-t** オプションを指定すると、**ranlib** コマンドはアーカイブをタッチするだけで修正しません。このコマンドは、アーカイブをコピーした後や、**ld** コマンドが古くなった記号テーブルに関してエラー・メッセージを表示しないように **make** コマンドの **-t** オプションを使用する場合に有効です。

## フラグ

**-t** 指定されたアーカイブを修正せずにタッチします。  
**-X mode** オブジェクト・ファイル **ranlib** のタイプを調べることを指定します。 *mode* には、次のいずれかを指定します。

**32** 32 ビットのオブジェクト・ファイルのみを処理します。

**64** 64 ビットのオブジェクト・ファイルのみを処理します。

**32\_64** 32 ビットと 64 ビットのオブジェクト・ファイルを処理します。

デフォルトでは、32 ビットのオブジェクトが処理されます (64 ビットのオブジェクト・ファイルは無視されます)。 *mode* は、**OBJECT\_MODE** 環境変数を使用して設定することもできます。例えば、**OBJECT\_MODE=64** を使用すると、**ranlib** は 64 ビット・オブジェクトをすべて処理し、32 ビット・オブジェクトは無視します。 **-X** フラグは **OBJECT\_MODE** 変数をオーバーライドします。

## 例

アーカイブ・ファイル `genlib.a` をランダム化するには、次のように入力します。

```
ranlib genlib.a
```

## ファイル

`/usr/ccs/bin/ranlib` **ranlib** コマンドが入っています。

## 関連情報

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」のサブルーチンの概要のセクション。

**ld** コマンド、 **ar** コマンド、 **lorder** コマンド、 **make** コマンド。

---

## raso コマンド

### 目的

RAS (Reliability (信頼性)、Availability (可用性)、Serviceability (保守性)) の各パラメーターを管理します。

## 構文

**raso** [-p | -r] [-y] [-o *Tunable* [= *Newvalue* ]

**raso** [-p | -r] [-y] [-d *Tunable*]

**raso** [-p] [-r] [-y] -D

**raso** [ -p ] [ -r ] -a

**raso** -h [ *Tunable* ]

**raso** -L [ *Tunable* ]

**raso** -x [ *Tunable* ]

注: 複数の **-o**、**-d**、**-x**、および **-L** フラグを指定できます。

## 説明

注: **raso** コマンドには、root 権限が必要です。

**raso** コマンドを使用して、RAS (信頼性・可用性・保守性) の各チューニング・パラメーターを構成します。**raso** コマンドは、すべての RAS チューニング・パラメーターの現行値または次のブート値を設定または表示します。**raso** コマンドを使用して、永続変更を行うことも、変更を次回リブートまで据え置くこともできます。指定されたフラグにより、**raso** コマンドがパラメーターを設定するのかまたは表示するのかを決定します。**-o** フラグを使用して、パラメーターの現在値を表示するか、またはパラメーターに新規の値を設定することができます。

## チューナブル・パラメーターの変更の影響を理解する

**raso** コマンドの誤用が原因で、パフォーマンスの低下またはオペレーティング・システムの障害が引き起こされる可能性があります。チューナブル・パラメーターを変更する前に、その目的を完全に理解するために、まず下記のチューナブル・パラメーターのセクションで、チューナブル・パラメーターの特性のすべてについて注意してお読みください。次に、このパラメーターの『診断』セクションおよび『チューニング』セクションに記載されている内容が、ご使用のシステムの状態に当てはまり、さらに、パラメーターの値を変更することが、システムのパフォーマンスの向上に役立つことを確認してください。診断セクションとチューニング・セクションの両方に「N/A」とのみ記されている場合は、AIX の開発側から特に指示がない限り、このパラメーターを変更しないことをお勧めします。

## フラグ

**-a**

すべてのチューナブル・パラメーターの現在値、リポート値 (**-r** フラグと併用した場合)、または永続値 (**-p** フラグと併用した場合) を表示します。*Tunable = Value* が組になり、1 行に 1 組ずつ表示されます。永続オプションでは、パラメーターのリポート値と現在値が等しい場合、パラメーターに対して 1 つの値が表示されるだけです。そうでない場合は、**NONE** が値として表示されます。

**-d** *Tunable*

*Tunable* をデフォルト値にリセットします。 *Tunable* を変更する必要がある場合 (すなわち、現在、デフォルト値に設定されていない) で、タイプが **Bosboot** または **Reboot** であるか、またはタイプが **Incremental** でデフォルト値が変更されており、**-r** フラグが併用されていない場合、

**-D**

*Tunable* の変更は行われず、警告が表示されます。すべてのチューナブル・パラメーターをそれぞれのデフォルト値にリセットします。変更の必要なチューナブル・パラメーターのタイプが **Bosboot** または **Reboot** である場合、あるいはタイプが **Incremental** で、デフォルト値から変更されており、**-r** が併用されていない場合、パラメーターは変更されず、警告が表示されます。

**-h** *Tunable*

*Tunable* パラメーターが指定されていない場合は、**raso** コマンドに関するヘルプを表示します。 *Tunable* パラメーターが指定されている場合は、*Tunable* パラメーターに関するヘルプを表示します。

**-L** *Tunable*

1 つまたはすべてのチューナブル・パラメーターの特性を、次の形式で 1 行に 1 つずつリストします。

| NAME               | CUR  | DEF  | BOOT | MIN | MAX  | UNIT      | TYPE |
|--------------------|------|------|------|-----|------|-----------|------|
| -----              |      |      |      |     |      |           |      |
| mtrc_commonbufsize | 3974 | 3974 | 3974 | 1   | 5067 | 4KB pages | D    |
| -----              |      |      |      |     |      |           |      |
| mtrc_enabled       | 1    | 1    | 1    | 0   | 1    | boolean   | B    |
| -----              |      |      |      |     |      |           |      |
| mtrc_rarebufsize   | 2649 | 2649 | 2649 | 1   | 3378 | 4KB pages | D    |
| -----              |      |      |      |     |      |           |      |

...  
 where:  
 CUR = current value  
 DEF = default value  
 BOOT = boot value  
 MIN = minimal value  
 MAX = maximum value  
 UNIT = tunable unit of measure  
 TYPE = parameter type: D (for Dynamic), S (for Static), R (for Reboot),  
 B (for Bosboot), M (for Mount), I (for Incremental) and C (for Connect)  
 DEPENDENCIES = list of dependent tunable parameters, one per line

**-o** *Tunable* [ =Newvalue ]

値を表示するか、*Tunable* を *Newvalue* に設定します。*Tunable* を変更する必要がある (指定された値が現在値と異なる)、そのタイプが **Bosboot** または **Reboot** である場合、あるいはタイプが **Incremental** で、現在値が指定された値より大きく、**-r** フラグが併用されていない場合、*Tunable* は変更されませんが警告が表示されます。

新しい値を指定せずに **-r** フラグを併用すると、*Tunable* の次のブート値が表示されます。新しい値を指定せずに **-p** フラグを併用すると、*Tunable* の現在値と次のブート値が同じ場合にのみ値が表示されます。そうでない場合は、**NONE** が値として表示されます。

**-p**

**-p** フラグを **-o**、**-d**、または **-D** フラグと併用すると、変更が現在値とリポート値の両方に適用されます (さらに、現在値が更新され、`/etc/tunables/nextboot` ファイルも更新されます)。Reboot および Bosboot タイプのパラメーターではこれらの組み合わせは使用できません。その理由は、これらのパラメーターの現在値を変更できないためです。

新しい値を指定せずに **-p** フラグを **-a** または **-o** フラグと併用した場合、パラメーターの現在値と次のブート値が同じ場合にのみ値が表示されます。そうでない場合は、**NONE** が値として表示されます。

**-r**

**-r** フラグを **-o**、**-d**、または **-D** フラグと併用すると、変更がリポート値に適用されます (`/etc/tunables/nextboot` ファイルが更新されます)。タイプが Bosboot のどのパラメーターを変更する場合も、**bosboot** コマンドを実行するようにプロンプトが出されます。

新しい値を指定せずに **-r** フラグを **-a** または **-o** フラグと併用すると、チューナブルの次回ブート時の値が現在値の代わりに表示されます。

**-y**

**bosboot** コマンドの実行前に、確認を求めるプロンプトの出力を抑制します。

## -x Tunable

1 つまたはすべてのチューナブルの特性を、以下の (スプレッドシート) 形式で、1 行に 1 つずつリストします。

Tunable Current Default Reboot Minimum Maximum Unit Type Dependencies

ここで、*Tunable* はチューナブル・パラメーター、*Current* はチューナブル・パラメーターの現在値、*Default* はチューナブル・パラメーターのデフォルト値、*Reboot* はチューナブル・パラメーターのリブート値です。また、*Minimum* はチューナブル・パラメーターの最小値、*Maximum* はチューナブル・パラメーターの最大値、*Unit* はチューナブル計測単位、*Type* はパラメーター・タイプ、*Dependencies* は従属チューナブル・パラメーターのリストです。

どのタイプ *Mount* のパラメーターを変更しても (-o、-d、または -D フラグを使用)、その変更は将来のマウントについてのみ有効であることを示す警告メッセージが表示されます。

どのタイプ *Connect* のパラメーターを変更しても (-o、-d、または -D フラグを使用)、*inetd* が再始動され、その変更は将来のソケット接続についてのみ有効であることを示す警告メッセージが表示されます。

-r フラグを無指定で、タイプ *Bosboot* または *Reboot* のパラメーターを変更 (-o、-d、または -D フラグ指定で) しようとするとき、エラー・メッセージが出されます。

*Incremental* タイプのパラメーターの現在値をそれより小さい新規の値に変更 (-o、-d、または -D フラグを使用しますが、-r フラグは使用しません) しようとするとき、エラー・メッセージが表示されます。

**bosboot** コマンドの実行前に、確認を求めるプロンプトの出力を抑制します。

## -y

## チューナブル・パラメーター

チューニング・コマンド (**no**、**nfso**、**vmo**、**ioo**、**schedo**、および **raso**) で取り扱われるすべてのチューナブル・パラメーターは、下記のカテゴリーに分類されています。

|             |                                                   |
|-------------|---------------------------------------------------|
| Dynamic     | パラメーターをいつでも変更できる場合                                |
| Static      | パラメーターをいかなる時にでも変更できない場合                           |
| Reboot      | パラメーターをリブート時にのみ変更できる場合                            |
| Bosboot     | <b>bosboot</b> を実行してマシンをリブートする場合にのみパラメーターを変更できる場合 |
| Mount       | パラメーターの変更が将来のファイルシステムまたはディレクトリーのマウントにのみ有効である場合    |
| Incremental | ブート時を除き、パラメーターを増やすことだけが可能な場合                      |
| Connect     | パラメーターの変更が将来のソケット接続にのみ有効である場合                     |

タイプ *Bosboot* のパラメーターの場合は、変更が加えられるたびに、チューニング・コマンドは自動的に、**bosboot** コマンドを実行したいかどうかを尋ねるプロンプトを出します。タイプ *Connect* のパラメーターの場合は、チューニング・コマンドは自動的に **inetd** デーモンを再始動します。

**schedo** コマンドが管理するパラメーターの現行セットには、Dynamic および Reboot タイプのみが含まれることに注意してください。

## 互換モード

5.2 より前の互換モードで実行する場合 (**sys0** の **pre520tune** 属性で制御される。パフォーマンス・マネージメント の **NFS tuning on the client** のセクションを参照)、パラメーターのリポート値は、タイプ **Bosboot** を除き、実際には意味がありません。このモードでは、値はブート時には適用されないからです。

5.2 より前の互換モードでは、チューニング・パラメーターへのリポート値の設定は、ブート中に呼び出されるスクリプト内でチューニング・コマンドを呼び出すことにより行います。したがって、タイプ **Reboot** のパラメーターは **-r** フラグなしで設定できるので、既存のスクリプトは従来どおり作動します。

このモードは、マシンを AIX 5.2 に移行すると、自動的にオンになります。完全なインストールの場合、これは **OFF** になり、パラメーターのリポート値は、リポート中に **/etc/tunables/nextboot** ファイルの内容を適用することにより設定されます。このモードの場合のみ、**-r** および **-p** フラグは完全に機能します。詳細については、*AIX 5L Version 5.3 Performance Tools Guide and Reference*のカーネル・チューニングのセクションを参照してください。

## チューナブル・パラメーター

kern\_heap\_noexec

**目的:** カーネル・ヒープに対して無実行保護を使用可能にするかどうかを指定します。保護を使用可能にした場合、カーネル・ヒープ内でコードを実行しようとする、カーネル例外となります。

**値:**

デフォルト: 0

範囲: 0 または 1

タイプ: Bosboot

**診断:** 該当しない

**チューニング:**

このチューナブルがサポートされるのは、64 ビットのカーネルおよび POWER4 以降のハードウェアのみです。

kernel\_noexec

**目的:** カーネル・データ領域に対して無実行保護を使用可能にするかどうかを指定します。保護を使用可能にした場合、保護領域内でコードを実行しようとする、カーネル例外となります。

**値:**

デフォルト: 1

範囲: 0 または 1

タイプ: Bosboot

**診断:** 該当しない

**チューニング:**

このチューナブルがサポートされるのは、64 ビットのカーネルおよび POWER4 以降のハードウェアのみです。

mbuf\_heap\_noexec

**目的:** mbuf ヒープに対して無実行保護を使用可能にするかどうかを指定します。保護を使用可能にした場合、mbuf ヒープ内でコードを実行しようとすると、カーネル例外となります。

**値:**

デフォルト: 0  
範囲: 0 または 1  
タイプ: Bosboot

**診断:** 該当しない

**チューニング:**

このチューナブルがサポートされるのは、64 ビットのカーネルおよび POWER4 以降のハードウェアのみです。

mtrc\_enabled

**目的:** Lightweight Memory Trace (LMT) 状態を定義します。

**値:**

デフォルト: 1  
範囲: 0 または 1  
タイプ: Bosboot

**診断:** 該当しない

**チューニング:**

値 1 は、LMT が使用可能であることを意味します。

mtrc\_commonbufsize

**目的:** Lightweight Memory Trace (LMT) の通常のイベントに関するメモリー・トレース・バッファー・サイズを指定します。LMT は、First Failure Data Capture (FFDC) にシステム・トレース情報を提供します。記録されたイベントは、システム・ダンプに保管されるか、またはユーザー・コマンドを介して報告されます (あるいは、このどちらも行われます)。

**値:**

デフォルト: 参照システム全体の活動状況およびハードウェアとシステム  
の特性のもとでのデータ生成に基づく値。  
範囲: 1 (4KB ページ・サイズ単位) からハードウェアとシステム  
の特性に基づく値まで。  
タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

ユーザー管理の動的バッファー・サイズの変更は、64 ビットのカーネル  
のみサポートされます。



mtc\_rarebufsize

**目的:** Lightweight Memory Trace (LMT) の、稀なイベントに関するメモリー・トレース・バッファ・サイズを指定します。LMT は、First Failure Data Capture (FFDC) にシステム・トレース情報を提供します。記録されたイベントは、システム・ダンプに保管されるか、またはユーザー・コマンドを介して報告されます (あるいは、このどちらも行われます)。

**値:**

デフォルト: 参照システム全体の活動状況およびハードウェアとシステム  
の特性のもとでのデータ生成に基づく値。

範囲: 1 (4KB ページ・サイズ単位) からハードウェアとシステム  
の特性に基づく値まで。

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

ユーザー管理の動的バッファ・サイズの変更は、64 ビットのカーネル  
のみサポートされます。

tprof\_cyc\_mult

**目的:** トレースのサンプリング頻度を制御する手段として Performance Monitor **PM\_CYC** およびソフトウェア・イベントのサンプリング頻度乗数を指定します。

**値:**

デフォルト: 1

範囲: 1 から 100

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

tprof\_evt\_mult

**目的:** トレース・サンプリング頻度を制御する手段として Performance Monitor **PM\_\*** イベント・サンプリング頻度乗数を指定します。

**値:**

デフォルト: 1

範囲: 1 から 10000

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**

該当しない

tprof\_inst\_threshold

**目的:** トレース・サンプリング頻度を制御する手段として Performance Monitor イベント・サンプル間の最小の完了指示数を指定します。

**値:**

デフォルト: 1000

範囲: 1 から 2G-1

タイプ: Dynamic

**診断:** 該当しない

**チューニング:**  
該当しない

## 例

1. **raso** コマンドが管理するすべてのチューナブル・パラメーターの、現在値とリポート値、範囲、単位、タイプ、および依存関係をリストするには、次のように入力します。

```
raso -L
```

2. Lightweight Memory Trace をオフにするには、次のように入力します。

```
raso -r -o mtrc_enabled=0
```

3. mtrc\_commonbufsize のヘルプを表示するには、次のように入力します。

```
raso -h mtrc_commonbufsize
```

4. 次のリポート後に tprof\_inst\_threshold を 10000 に設定するには、次のように入力します。

```
raso -r -o tprof_inst_threshold=10000
```

5. すべての **raso** チューナブル・パラメーターをそのデフォルト値に永続的にリセットするには、次のように入力します。

```
raso -p -D
```

6. すべての Virtual Memory Manager チューニング・パラメーターのリポート・レベルをリストするには、次のように入力します。

```
raso -r -a
```

7. **raso** コマンドが管理するすべてのチューナブル・パラメーターの、現在値とリポート値、範囲、単位、タイプ、および依存関係をリスト (スプレッドシート形式) するには、次のように入力します。

```
raso -x
```

## 関連情報

**ioo** コマンド、**nfso** コマンド、**no** コマンド、**schedo** コマンド、**tunchange** コマンド、**tuncheck** コマンド、**tundefault** コマンド、**tunrestore** コマンド、**tunsave** コマンド、**vmo** コマンド。

*AIX 5L Version 5.3 Performance Tools Guide and Reference* の Use of the tprof program to analyze programs for CPU use のセクション。

---

## ras\_logger コマンド

### 目的

エラー・テンプレートを使用してエラーをログに記録します。

## 構文

`/usr/lib/ras/ras_logger [ -y template-file ]`

## 説明

**ras\_logger** コマンドは、データをログに記録する方法を決めるエラーのテンプレートを使用して、標準入力に提供された 1 つのエラーをログに記録します。入力のフォーマットを以下に示します。

```
error_label
resource_name
64_bit_flag
detail_data_item1
detail_data_item2
...
```

**error\_label** フィールドは、テンプレートで定義されたエラーのラベルです。 **resource\_name** フィールドの長さは 16 文字までです。 **64\_bit\_flag** フィールドの値は、32 ビット・エラーの場合は 0、64 ビット・エラーの場合は 1 です。 **detail\_data** フィールドは、テンプレート内の **Detail\_Data** 項目に対応します。

## フラグ

`-y template-file`

`/var/adm/ras/errtmpl` デフォルト・ファイル以外のテンプレート・ファイルを指定します。

## 例

1. エラーをログに記録します。テンプレートは以下のとおりです。

```
+ F00:
Catname = "foo.cat"
Err_Type = TEMP
Class = 0
Report = TRUE
Log = TRUE
Alert = FALSE
Err_Desc = {1, 1, "Error F00"}
Prob_Causes = {1, 2, "Just a test"}
User_Causes = {1, 2, "Just a test"}
User_Actions = {1, 3, "Do nothing"}
Detail_Data = 4, {2, 1, "decimal"},DEC
Detail_Data = W, {2, 1, "hex data"},HEX
Detail_Data = 100, {2, 1, "long string"},ALPHA
```

**tfile** ファイル内の `ras_logger` 入力は以下のとおりです。

```
F00
resource
0
15
A0
hello world
```

`/usr/lib/ras/ras_logger <tfile` コマンドを実行します。これは、リソース名が **resource** である F00 エラーをログに記録します。詳細データは、decimal 15 にセットされた 4 バイト、0xa0 にセットされた 16 進データの 4 バイト、および文字列 "hello world" からなります。64 ビット・フラグの値が 1 の場合、16 進データは 8 バイトが 0xa0 にセットされます。

2. 複数項目の 10 進数値です。テンプレートは以下のとおりです。

```
+ F00:
  Catname = "foo.cat"
  Err_Type = TEMP
  Class = 0
  Report = TRUE
  Log = TRUE
  Alert = FALSE
  Err_Desc = {1, 1, "Error F00"}
  Prob_Causes = {1, 2, "Just a test"}
  User_Causes = {1, 2, "Just a test"}
  User_Actions = {1, 3, "Do nothing"}
  Detail_Data = 8, {2, 1, "decimal"}, DEC
  Detail_Data = W, {2, 1, "hex data"}, HEX
  Detail_Data = 100, {2, 1, "long string"}, ALPHA
```

**ras\_logger** コマンドは、**tfile** ファイルにある次の情報を入力とします。

```
F00
resource
0
15 -15
A0
hello world
```

注: 10 進データは通常 **errpt** コマンドにより、2 つの別の値 (各 4 バイト) として示されます。したがって、入力は 15 および -15 を含みます。これが **errpt** コマンドの表示方法です。

## 関連情報

**errpt** および **errlogger** コマンド。

---

## rc コマンド

### 目的

標準の始動初期化を行います。

### 構文

**rc**

### 説明

**rc** コマンドは、**/etc/inittab** ファイルにエントリを持っています。 **init** コマンドは、**/etc/inittab** ファイル内の **rc** コマンドについてプロセスを作成します。 **rc** コマンドは、システムの標準の起動初期化を行います。 **/etc/rc** の内容はインストール時特有です。必要なオペレーションがすべて正常に完了すると、**init** コマンドがロガーを開始して標準の初期化と始動を完了できるように、ゼロ戻りコードを表示してファイルが終了します。

注:

1. ページ・スペースの活動化およびファイルシステムのマウントなどの数多くの立ち上げ機能を **rc** コマンドで実行します。
2. ルート・ファイルシステムは暗黙指定によってマウントされます。

## 関連情報

**fsck** コマンド、**init** コマンド、**mount** コマンド。

---

## rc.mobip6 コマンド

### 目的

システムが、モバイル IPv6 ホーム・エージェントまたは対応するノードとして機能できるようにします。

### 構文

```
rc.mobip6 { start [ -H ] [ -S ] | stop [ -N ] [ -F ] }
```

### 説明

**/etc/rc.mobip6** ファイルはシェル・スクリプトであり、実行すると、システムがモバイル IPv6 ホーム・エージェントまたは対応するノードとして機能できるようにします。システム管理を使用して、システムを再始動するたびにモバイル IPv6 を開始するように構成した場合、スクリプトは再始動時に自動的に実行されます。

### フラグ

|           |                                                                                          |
|-----------|------------------------------------------------------------------------------------------|
| <b>-F</b> | IPv6 の転送を使用不可にします。                                                                       |
| <b>-H</b> | システムを、モバイル IPv6 ホーム・エージェントおよび対応するノードとして使用可能にします。このフラグを指定しないと、システムは対応するノードとしてのみ使用可能になります。 |
| <b>-N</b> | <b>ndpd-router</b> デーモンを停止します。                                                           |
| <b>-S</b> | IP セキュリティー認証の検査を使用可能にします。                                                                |

### 終了状況

- 0 コマンドは正常に実行されました。
- >0 エラーが発生しました。

### セキュリティ

このコマンドを実行するには、root 権限を持つか、または system グループのメンバーである必要があります。

### 例

- 次の例は、システムをモバイル IPv6 ホーム・エージェントおよび対応するノードとして使用可能にします。  

```
/etc/rc.mobip6 start -H
```
- 次の例は、システムをモバイル IPv6 対応ノードとして使用可能にし、IP セキュリティー検査を使用可能にします。  

```
/etc/rc.mobip6 start -S
```
- 次の例は、システムのすべてのモバイル IPv6 および IPv6 ゲートウェイ機能を使用不可にします。  

```
/etc/rc.mobip6 stop -N -F
```
- 次の例は、システムのすべてのモバイル IPv6 機能を使用不可にするけれども、IPv6 ゲートウェイとしての機能は続行できるようにします。  

```
/etc/rc.mobip6 stop
```

# ファイル

/etc/rc.mobip6

rc.mobip6 コマンドが入っています。

## 関連情報

kmodctrl コマンド、mobip6reqd コマンド、mobip6ctrl コマンド、ndpd-router コマンド。

ネットワークおよびコミュニケーションのマネージ のモバイル IPv6 のセクション。

---

## rc.powerfail コマンド

### 目的

RPA (RS/6000 プラットフォーム・アーキテクチャー) に特有の EPOW (Environmental and Power Warning) イベントを処理し、EPOW イベント処理の一部として必要に応じてシステムをシャットダウンします。

### 構文

```
rc.powerfail [ -h ] | [ [ -s ] [ -t [ mm ] ] [-c [ ss ] ] ]
```

### 説明

rc.powerfail コマンドは、init がカーネルから SIGPWR シグナルを受信した場合に、/etc/inittab ファイルによって起動されます。rc.powerfail コマンドは、ioctl() を使用することによってシステムの状態を判別します。rc.powerfail コマンドを呼び出すのは、EPOW イベントの発生時だけにしてください。

rc.powerfail によって処理されるさまざまな EPOW イベントと、それに対応する rc.powerfail によるイベント処理を、次の表に示します。

| EPOW クラス                                                      | rc.powerfail によるイベント処理                                                                                         | 例                       |
|---------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|-------------------------|
| 1<br>これらのタイプのエラーは、オペレーティング・システムにとっては、あまり重要でない冷却の問題であると見なされます。 | rc.powerfail は、現在システムにログオンしているユーザーに対し、cron エントリーによって警告します。それは、この状態が解決されるまで 12 時間ごとに出現します。                      | 予備ファンの障害。内部の熱の問題。       |
| 2<br>これらのタイプのエラーは、オペレーティング・システムにとっては、あまり重要でない電源の問題であると見なされます。 | rc.powerfail は、現在システムにログオンしているユーザーに対し、cron エントリーによって警告します。それは、この状態が解決されるまで 12 時間ごとに出現します。                      | 予備 AC 入力障害。             |
| 3<br>これらのイベントは本質的に重大なものであり、可能な限り速やかにシステムの電源をオフにしなければなりません。    | rc.powerfail は、-t オプションで別の待ち時間値をユーザーが指定したのでない限り、10 分以内にシステム・シャットダウンを開始します。                                     | 周囲の温度が所定の限界値に達しつつある場合。  |
| 4<br>この種類のエラーは本質的に緊急事態であり、システムをすぐに停止する必要があります。                | rc.powerfail は、このイベントを 20 秒以内に処理することが预期されています。このような場合、rc.powerfail は、現在システムにログオンしているユーザーに警告した後、ただちにシステムを停止します。 | AC 電源障害。どの電源の電力も失われた場合。 |

| EPOW クラス                                           | rc.powerfail によるイベント処理                                                                                                         | 例                                      |
|----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|
| 5、7<br>この種類のエラーは本質的に緊急事態であり、マイクロ秒の単位で対応しなければなりません。 | 数マイクロ秒以内に対応しなければならないため、これらのイベントは <b>rc.powerfail</b> によっては処理されません。このような状態で <b>rc.powerfail</b> が制御を得た場合、それは待ち時間が終了するまで待機し続けます。 | ファン・システムのすべてに障害が発生し、予備でない電源に障害が発生した場合。 |

前述のように EPOW クラス 3 のイベントの場合、**rc.powerfail** コマンドがシステムをシャットダウンするまでに約 10 分の時間があります。その時間の長さは、ユーザーが **/etc/inittab** ファイルの **powerfail** エントリーで **-t** オプションを使用することによって変更できます。最後の 60 秒間に突入する前に、まだログオンしているユーザーに対して、シャットダウンまでの残り時間を示すメッセージが送られます。最後の 60 秒間のどこかでイベントがクリアされると、システムのシャットダウンは停止し、すべてのエラーがクリアされたことがユーザーに通知されます。シャットダウンが望ましくない場合、ユーザーは **/etc/inittab** ファイルの中のこのコマンドに **-s** オプションを追加できます。

また、EPOW クラス 3 のイベントの場合、**rc.powerfail** は、環境特有のスクリプトがあるなら、システム・シャットダウンの前にそれらが実行されるようにします。それらのスクリプトは **/usr/lib/epow/scripts** に入っており、**rc.powerfail** は、それらが完了するまでデフォルトで 10 秒間待機します。この待ち時間は **-c** オプションを使用して変更できます。**-c** オプションによって指定される値は、それらのスクリプトのための待ち時間の秒数と解釈されます。

## フラグ

- h**      パワー状況コードおよびその処置が入っている通知メッセージを指定します。**rc.powerfail -h** コマンドは、EPOW イベント処理の一部として必要に応じてシステムをシャットダウンします。
- s**      バッテリー・バックアップまたはファンの故障のいずれかによるシステムの電源障害が生じた場合に、システムのシャットダウンを実行しません。ログオン・ユーザーは適切なメッセージをすべて受信しますが、実際のシステムのシャットダウンはシステム管理者にゆだねられます。重大な電源障害が検出された場合には、このフラグは効力を持ちません。
- t mm**    バッテリー・バックアップまたはファンの故障による主電源の消失の場合の、システムのシャットダウンまでの合計時間の分数を指定します。この数値はバッテリー・バックアップが保証する時間の半分に等しくなければなりません。重大な電源障害が検出された場合には、このフラグは効力を持ちません。
- c ss**    EPOW 3 の状況で、環境特有のサード・パーティー製スクリプトの実行完了を **rc.powerfail** が待機する時間を指定します (秒単位)。

## 終了状況

システムがシャットダウンした場合は、終了値は戻されません。それ以外の場合は、**rc.powerfail** コマンドによって以下の終了値が戻されます。

- 0          正常な状態。
- 1          構文エラー。
- 2          **halt -q** が失敗しました。
- 3          **shutdown -F** が失敗しました。
- 4          エラーが発生しました。**shutdown -F** を使用してただちにシステムをシャットダウンしてください。
- 5          不定状態。サービス技術員に連絡してください。

## セキュリティ

アクセス制御: ルートのみ。



## 例

1. 電源状況が 3 の原因を調べるには、次のように入力します。

```
rc.powerfail -h
```

2. クリティカルでない電源障害またはファンの故障が発生した場合に、システムのシャットダウンを防止するには、次のように入力します。

```
chitab "powerfail::powerfail:/etc/rc.powerfail -s >dev/console 2>&1"
```

次に **init** が受信する SIGPWR は、クリティカルでない電源障害が発生した場合に、システムのシャットダウンを実行しません。

3. シャットダウンまでの時間を 30 分に変更するには、次のように入力します。

```
chitab "powerfail::powerfail:/etc/rc.powerfail -t 30 >/dev/console 2>&1"
```

条件が重大でない限り、**init** が次に受信する SIGPWR の遅延は、システム・シャットダウンまで 30 分になります。

## ファイル

**/etc/inittab**                    初期化プロセスを制御します。

## 関連情報

**machstat** コマンド、**init** コマンド、**chitab** コマンド、**halt** コマンド、**shutdown** コマンド。

---

## rcp コマンド

### 目的

ローカル・ホストとリモート・ホスト間または 2 つのリモート・ホスト間でファイルを転送します。

### 構文

```
rcp [ -p] [ -F] [ -k realm ] [ -m] { { User@Host:File | Host:File | File } { User@Host:File | Host:File | File | User@Host:Directory | Host:Directory | Directory } | [ -r] { User@Host:Directory | Host:Directory | Directory } }
```

### 説明

**/usr/bin/rcp** コマンドは、ローカル・ホストとリモート・ホスト間、2 つのリモート・ホスト間、または同じリモート・ホスト上のファイル間で、単一または複数のファイルをコピーするときに使用します。

リモート宛先ファイルとディレクトリーについては、**Host:** パラメーターを指定する必要があります。ソースまたは宛先のどちらにもリモート・ホスト名を指定しない場合は、**rcp** コマンドは **cp** コマンドと等価になります。ローカル・ファイル名およびディレクトリー名に **Host:** パラメーターを指定する必要はありません。

**注:** **rcp** コマンドでは、**:** (コロン) はホスト名の終わりを示すものであると想定します。ファイル名に **:** を使用する場合には、ファイル名の前に **/** (スラッシュ) を付けるか、または **/** を含む絶対パス名で使用してください。

*Host* の前に接頭部として *User@* パラメーターを付けないと、リモート・ホストではローカル・ユーザー名が使用されます。*User@* パラメーターを入力すると、その名前が使われます。

リモート・ホスト上のファイルまたはディレクトリーに対してパスを指定しないか、完全に修飾しなければ、パスがリモート・ユーザー・アカウント用のホーム・ディレクトリーから開始すると解釈されます。また、リモート・ホストで解釈しなければならないメタキャラクターは、\ (バックスラッシュ)、" (二重引用符)、または ' (単一引用符) で囲まなければなりません。

## ファイル許可と所有権

デフォルトでは、既存の宛先ファイルの許可モードと所有権は保存されます。通常、宛先ファイルが存在しなければ、宛先ファイルの許可モードは宛先ホスト側で **umask** コマンド (Korn シェルの特殊コマンド) によって修正されるソース・ファイルの許可モードに等しくなります。**rcp** コマンドの **-p** フラグを設定すると、ソース・ファイルの修正時刻とモードが宛先ホスト側に保存されます。

リモート・ホスト用に入力するユーザー名によって、**rcp** コマンドがそのホストで使用するファイル・アクセス特権が決まります。また、宛先ホストに与えられたユーザー名によって、結果的に生成される宛先ファイルの所有権とアクセス・モードが決まります。

## 標準の認証の使用

リモート・ホストには、以下の条件のいずれかが満たされるとアクセスできます。

- ローカル・ホストがリモート・ホストの **/etc/hosts.equiv** ファイルに組み込まれていて、リモート・ユーザーが **root** ユーザーでない場合。
- ローカル・ホストおよびユーザー名がリモート・ユーザーのアカウントにある **\$HOME/.rhosts** ファイルに組み込まれている場合。

ユーザーは **\$HOME/.rhosts** ファイルに対して、許可を任意に設定できますが、**.rhosts** ファイルの許可は 600 (オーナーだけが読み取りおよび書き込み可能) に設定することをお勧めします。

上記の条件のほかに、リモート・ユーザーのアカウントにパスワードが定義されていなければ、**rcp** コマンドを使用してリモート・ホストにアクセスすることができます。ただし、セキュリティ上の理由で、すべてのユーザー・アカウント上でパスワードを使用することをお勧めします。

## Kerberos 5 認証の場合

次の条件のすべてが満たされた場合にのみ、リモート・ホストへのアクセスが可能になります。

- ローカル・ユーザーに現在の DCE 証明書が与えられている。
- ローカル・システムとリモート・システムが、Kerberos 5 認証に対して構成されている (一部のリモート・システムの場合、これは不要のことがある。デーモンが **klogin** ポートを **listen** していることが必要)。
- リモート・システムが DCE 証明書をリモート・アカウントへのアクセスに充分であるとして受け入れる。追加情報については、**kvalid\_user** 関数を参照のこと。

## rcp および名前付きパイプライン

名前付きパイプライン、または FIFO (**mknod -p** コマンドで作成するスペシャル・ファイル) をコピーするときには、**rcp** コマンドを使用しないでください。**rcp** コマンドは、コピーの対象となるファイルに **open** サブルーチンを使用しますが、このサブルーチンは FIFO パイプ接続のようなブロック化デバイスはブロックします。

## 制約事項

SP Kerberos V4 rcp 実行パスは、Kerberos がクレデンシャルの転送をサポートしないため、リモート間コピーをサポートしません。このような状況下で受け取るメッセージは、ユーザーがチケットを持っていないことを示すメッセージなので、**kinit** を使用してログインする必要があります。メッセージを出すのは、リモート・ソース・マシンです。以下の Kerberos を使用する場合の例を参照して、リモート間コピーを行ってください。

## フラグ

- p** ユーザーが root 権限を持っているか、または宛先のオーナーである場合に限り、宛先に送るコピー内にソース・ファイルの修正時刻とモードを保存します。このフラグを使用しないと、宛先側の **umask** コマンドが宛先ファイルのモードを修正し、宛先ファイルの修正時刻はそのファイルが受け取られた時刻に設定されます。  
  
このフラグを使用しないと、有効な **umask** 値が適切なデータベースに保存されます。この値は **umask** コマンドを発行することによって設定される値とは異なります。**umask** コマンドで指定された許可と所有権の値は、データベース内に保存された値には影響しません。
- r** ソース・ディレクトリー内の各ファイルとサブディレクトリーを、宛先ディレクトリーに再帰コピーします。
- F** これを指定すると、証明書が転送されます。さらに、リモート・システムの証明書に転送可能のマークが付けられます (これによって、別のリモート・システムにそれらの証明書を渡すことができるようになります)。このフラグは、Kerberos 5 が現在の認証メソッドでない場合は無視されます。認証は、現在の DCE 証明書に転送可能のマークが付けられていない場合には無視されます。
- k realm** これを使用することで、ユーザーは、リモート端末のレルムを指定できます。ただし、そのレルムが DCE セルと同意の場合に限りです。このフラグは、Kerberos 5 が現在の認証メソッドでない場合は無視されます。
- m** ファイル名のメタキャラクターのサポート。

## セキュリティ

次の条件のうち、少なくとも 1 つが満たされた場合にのみ、リモート・ホストへのアクセスが可能になります。

- ローカル・ユーザー ID が認証データベース内でプリンシパルとしてリストされ、**kinit** を実行してチケットを獲得していた場合。
- **\$HOME/.klogin** ファイルがある場合は、そのファイルがターゲット・システム上のローカル・ユーザーの **\$HOME** ディレクトリーにある場合。ローカル・ユーザーは、さらにこのアカウントに **rsh** を許可されたユーザーまたはサービスとしてリストされなければなりません。このファイルは、ローカル **rhosts** ファイルと同等の機能を実行します。このファイル内の各行には、“principal.instance@realm” 形式のプリンシパルがなければなりません。親ユーザーが **.klogin** 内で命名されたプリンシパルの 1 つとして認証されている場合は、アカウントへのアクセスが認可されます。アカウントのオーナーは **.klogin** ファイルがない場合にアクセスが認可されます。

セキュリティ上の理由から、**\$HOME/.klogin** ファイルはリモート・ユーザーが所有し、AIX 所有者 ID のみが **klogin** への読み取り/書き込みアクセス権限 (許可 = 600) を持つようにします。

## パラメーター

|                            |                                                                                                                                                                                                                |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Host:File</i>           | : (コロン) で区切られたリモート宛先ファイルのホスト名 ( <i>Host</i> ) およびファイル名 ( <i>File</i> ) です。<br><b>注:</b> <b>rcp</b> コマンドでは : (コロン) がホスト名の終わりを表すものと想定されるため、ローカル・ファイル名またはディレクトリー名に組み込まれているコロンの前にはすべて \ (バックスラッシュ) を挿入しなければなりません。  |
| <i>User@Host:File</i>      | <b>rcp</b> コマンドが転送したファイルの所有権を設定するときに使用するユーザー名 ( <i>User@</i> )、ホスト名 ( <i>Host</i> )、およびリモート宛先ファイルのファイル名 ( <i>File</i> ) を指定します。リモート・ホスト用に入力するユーザー名によって、 <b>rcp</b> コマンドがそのホストで使用するファイル・アクセス特権が決まります。           |
| <i>File</i>                | ローカル宛先ファイルのファイル名を指定します。                                                                                                                                                                                        |
| <i>Host:Directory</i>      | リモート宛先ディレクトリーのホスト名 ( <i>Host</i> ) およびディレクトリー名 ( <i>Directory</i> ) を指定します。<br><b>注:</b> <b>rcp</b> コマンドでは : (コロン) がホスト名の終わりを表すものと想定されるため、ローカル・ファイル名またはディレクトリー名に組み込まれているコロンの前にはすべて \ (バックスラッシュ) を挿入しなければなりません。 |
| <i>User@Host:Directory</i> | <b>rcp</b> コマンドが転送したファイルの所有権を設定するときに使用するユーザー名 ( <i>User@</i> )、ホスト名 ( <i>Host</i> )、およびディレクトリー名 ( <i>Directory</i> ) を指定します。リモート・ホスト用に入力するユーザー名によって、 <b>rcp</b> コマンドがそのホストで使用するファイル・アクセス特権が決まります。              |
| <i>Directory</i>           | ローカル宛先ディレクトリーのディレクトリー名。                                                                                                                                                                                        |

## 終了状況

このコマンドは、以下の終了値を戻します。

|    |             |
|----|-------------|
| 0  | 正常終了。       |
| >0 | エラーが発生しました。 |

## 例

以下の例では、ローカル・ホストはリモート・ホストにある **/etc/hosts.equiv** ファイルでリストされます。

1. ローカル・ファイルをリモート・ホストにコピーするには、以下のように入力します。

```
rcp localfile host2:/home/eng/jane
```

ローカル・ホストからのファイル *localfile* がリモート・ホスト *host2* にコピーされます。

2. あるリモート・ホストから別のリモート・ホストにリモート・ファイルをコピーするには、以下のように入力します。

```
rcp host1:/home/eng/jane/newplan host2:/home/eng/mary
```

ファイル */home/eng/jane/newplan* がリモート・ホスト *host1* からリモート・ホスト *host2* にコピーされます。

3. ローカル・ホストからのディレクトリー・サブツリーをリモート・ホストに送信して、修正時刻とモードを保存するには、以下のように入力します。

```
rcp -p -r report jane@host2:report
```

ディレクトリー・サブツリー `report` が、ローカル・ホストからリモート・ホスト `host2` のユーザー `jane` のホーム・ディレクトリーにコピーされ、すべてのモードと修正時刻が保存されます。リモート・ファイル `/home/jane/rhosts` には、ローカル・ホストおよびユーザー名を指定するエントリーが含まれます。

- この例は、ターゲットとサーバーの両方で認証が Kerberos 4 の場合に、root ユーザーがリモート・ホスト上で `rcp` を出す方法を示しています。root ユーザーは認証データベースに登録され、ローカル・ホスト上に `kinit` を出しておく必要があります。コマンドはローカル・ホストで出され、ファイル `stuff` をノード `r05n07` から SP 上の `r05n05` へコピーします。

```
/usr/lpp/ssp/rcmd/bin/rsh r05n07 'export KRBTKFILE=/tmp/rcmdtkkt$$; \  
/usr/lpp/ssp/rcmd/bin/rcmdtgt; \  
/usr/lpp/ssp/rcmd/bin/rcp /tmp/stuff r05n05:/tmp/stuff;'
```

root ユーザーは、`KRBTKFILE` 環境変数を一時チケット・キャッシュ・ファイル名に設定してから、`rcmdtgt` コマンドを出してサービス・チケットを獲得します。`rcp` はサービス・チケットを使用して、ホスト `r05n07` からホスト `r05n05` へ認証します。

## ファイル

|                                        |                                                                                              |
|----------------------------------------|----------------------------------------------------------------------------------------------|
| <code>\$/HOME/.klogin</code>           | ローカル・ユーザー・アカウントを使用できるリモート・ユーザーを指定します。                                                        |
| <code>/usr/lpp/ssp/rcmd/bin/rcp</code> | SP Kerberos 4 <code>rcp</code> ルーチンを (必要な場合に) 呼び出す AIX セキュア <code>/usr/bin/rsh</code> へのリンク。 |

## 前提条件情報

概要については、「IBM Parallel System Support Programs for AIX: Administration Guide」のセキュリティに関する章を参照してください。この資料には、Web サイト [http://www.rs6000.ibm.com/resource/aix\\_resource](http://www.rs6000.ibm.com/resource/aix_resource) でアクセスできます。

Kerberos の追加情報については、「IBM Parallel System Support Programs for AIX: Command and Technical Reference」の RS/6000 SP Files and Other Technical Information のセクションを参照してください。この資料には、Web サイト [http://www.rs6000.ibm.com/resource/aix\\_resource](http://www.rs6000.ibm.com/resource/aix_resource) でアクセスできます。

## 関連情報

`cp` コマンド、`ftp` コマンド、`rlogin` コマンド、`rsh` コマンド、`telnet` コマンド、`tftp` コマンド、`umask` コマンド。

SP コマンド: `k4init`、`k4list`、`k4destroy`、`lsauthpar`、`chauthpar`、`Kerberos`。

環境変数 `KRBTKFILE`。

`rshd` および `krshd` デーモン。

`kvalid_user` 関数。

`.rhosts` ファイル・フォーマット、`hosts.equiv` ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージの通信およびネットワークのセクション。

ネットワークおよびコミュニケーションのマネージの認証およびセキュア `rcmd` のセクション。

---

## rcvdist コマンド

### 目的

着信メッセージのコピーを追加の宛先に送信します。

### 構文

```
rcvdist [ -form File ] User ...
```

### 説明

**rcvdist** コマンドは、着信メッセージのコピーを本来の受信者以外のユーザーに送ります。**rcvdist** コマンドを開始するのは、ユーザーではありません。**rcvdist** コマンドは、**/usr/lib/mh/slocal** コマンドが呼び出す **.maildelivery** ファイル内に登録されます。

**rcvdist** コマンドは、*User* パラメーターで指定されたユーザーに着信メッセージのコピーを送信します。デフォルト文字列は、**rcvdistcomps** ファイルにあります。このファイルは、コマンドからの出力をフォーマットし、**send** コマンドを通じて、指定された ID または別名に送ります。

**rcvdistcomps** ファイルをローカル・メール・ディレクトリーの中にコピーして、必要に応じて文字列を変更できます。メッセージ・ハンドラー (MH) パッケージは、まず、ローカル・メール・ディレクトリー内の **rcvdistcomps** ファイルを使用します。あるいは、**-form** フラグを使用して、希望する文字列が入っているファイル名を指定することもできます。

### フラグ

- form File** コマンド出力をフォーマット設定するファイルを指定します。デフォルトは、**rcvdistcomps** ファイルです。
- help** コマンド構文、使用可能なスイッチ (トグル)、およびバージョン情報をリストします。  
注: MH の場合は、このフラグ名を省略することはできません。

### ファイル

- \$HOME/.maildelivery** ローカル・メール送達のための MH 命令を提供します。
- \$HOME/forward** デフォルトのメッセージ・フィルターを提供します。

### 関連情報

**ali** コマンド、**rcvpack** コマンド、**rcvstore** コマンド、**rcvttty** コマンド、**sendmail** コマンド、**slocal** コマンド、**whom** コマンド。

**.mh\_alias** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ のメール・アプリケーションのセクション。



---

## rcvpack コマンド

### 目的

着信メッセージをパック・ファイルに保存します。

### 構文

**rcvpack** [ *File* ]

### 説明

**rcvpack** コマンドは、着信メッセージを *File* パラメーターで指定されたパック・ファイル内に配置します。**rcvpack** コマンドを始動させるのは、ユーザーではありません。**rcvpack** コマンドは、**/usr/lib/mh/slocal** コマンドが呼び出す **\$HOME/.mailedelivery** ファイル内に登録されます。

**rcvpack** コマンドは、指定されたファイルに着信メッセージのコピーを追加し、そのファイルで **packf** コマンドを実行します。**.mailedelivery** ファイルで **rcvpack** コマンドを指定すると、すべての着信メッセージで **rcvpack** コマンドを実行します。

### フラグ

**-help** コマンド構文、使用可能なスイッチ (トグル)、およびバージョン情報をリストします。  
注: MH の場合は、このフラグ名を省略することはできません。

### ファイル

|                              |                             |
|------------------------------|-----------------------------|
| <b>\$HOME/.mailedelivery</b> | ローカル・メール送達のための MH 命令を提供します。 |
| <b>\$HOME/.forward</b>       | デフォルトのメッセージ・フィルタを提供します。     |

### 関連情報

**inc** コマンド、**packf** コマンド、**rcvdist** コマンド、**rcvstore** コマンド、**rcvttty** コマンド、**sendmail** コマンド、**slocal** コマンド。

**.mh\_alias** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ のメール・アプリケーションのセクション。

---

## rcvstore コマンド

### 目的

標準入力からの新規メールをフォルダーに取り込みます。

### 構文

**rcvstore** [ *+Folder* ] [ **-create** | **-nocreate** ] [ **-sequence** *Name* ] [ **-public** | **-npublic** ] [ **-zero** | **-nozero** ]



## 説明

**rcvstore** コマンドは、着信メッセージを指定された **message** ディレクトリー (フォルダー) に追加します。**rcvstore** コマンドを始動させるのは、ユーザーではありません。**rcvstore** コマンドは、**/usr/lib/mh/slocal** コマンドが呼び出す **\$HOME/.maildelivery** ファイル内に登録されます。

**rcvstore** は、標準入力からのメッセージを受け入れて、指定されたフォルダー内に配置します。**.maildelivery** ファイル内に **rcvstore** コマンドを指定すると、すべての受信メッセージについて **rcvstore** コマンドが実行されます。

**rcvstore** コマンド・フラグは、**\$HOME/.maildelivery** ファイル内で指定するか、ほとんどの MH コマンドと同様に **\$HOME/.mh\_profile** 内で指定できます。

## フラグ

|                       |                                                                                                                                                 |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-create</b>        | フォルダーが存在しない場合は、指定されたフォルダーをメール・ディレクトリーに作成します。このフラグがデフォルトです。                                                                                      |
| <b>+Folder</b>        | 組み込まれたメッセージを指定されたフォルダー内に配置します。デフォルトは、 <b>+inbox</b> です。                                                                                         |
| <b>-help</b>          | コマンド構文、使用可能なスイッチ (トグル)、およびバージョン情報をリストします。<br><b>注:</b> MH の場合は、このフラグ名を省略することはできません。                                                             |
| <b>-nocreate</b>      | フォルダーが存在しない場合、指定されたフォルダーを作成しません。                                                                                                                |
| <b>-nopublic</b>      | 指定されたメッセージ・シーケンスをユーザーが使用することを制限します。 <b>-nopublic</b> フラグは、シーケンス内のメッセージを制限するのではなく、シーケンスのみを制限します。フォルダーがほかのユーザーに対して書き込み保護されている場合は、このフラグがデフォルトです。   |
| <b>-nozero</b>        | <b>rcvstore</b> コマンドが組み込んだメッセージを指定されたメッセージ・シーケンスに追加します。このフラグがデフォルトです。                                                                           |
| <b>-public</b>        | 指定されたメッセージ・シーケンスをほかのユーザーが使用できるようにします。 <b>-public</b> フラグは、保護メッセージを使用可能にするのではなく、シーケンスのみを使用可能にします。フォルダーがほかのユーザーに対して書き込み保護されていない場合は、このフラグがデフォルトです。 |
| <b>-sequence Name</b> | 組み込まれたメッセージを <b>Name</b> パラメーターで指定されたメッセージのシーケンスに追加します。                                                                                         |
| <b>-zero</b>          | 指定されたメッセージ・シーケンスをクリアしてから、組み込まれたメッセージをシーケンスに配置します。このフラグがデフォルトです。                                                                                 |

## プロファイル・エントリー

|                         |                                                     |
|-------------------------|-----------------------------------------------------|
| <b>Folder-Protect:</b>  | 新規フォルダー・ディレクトリーの保護レベルを設定します。                        |
| <b>Msg-Protect:</b>     | 新しいメッセージ・ファイルの保護レベルを設定します。                          |
| <b>Path:</b>            | <b>UserMHDIRECTORY</b> (ユーザーの MH ディレクトリー) 変数を指定します。 |
| <b>Unseen-Sequence:</b> | ユーザーが未読のメッセージのトラックに使うコマンド・シーケンスを指定します。              |
| <b>Rcvstore:</b>        | <b>rcvstore</b> プログラムのフラグを指定します。                    |

## ファイル

|                             |                             |
|-----------------------------|-----------------------------|
| <b>\$HOME/.mh_profile</b>   | MH ユーザー・プロファイルが入っています。      |
| <b>\$HOME/.maildelivery</b> | ローカル・メール送達のための MH 命令を提供します。 |
| <b>\$HOME/.forward</b>      | デフォルトのメッセージ・フィルターを提供します。    |

## 関連情報

**inc** コマンド、 **rcvdist** コマンド、 **rcvpack** コマンド、 **rcvttty** コマンド、 **sendmail** コマンド、 **slocal** コマンド。

**.mh\_alias** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ のメール・アプリケーションのセクション。

---

## rcvttty コマンド

### 目的

ユーザーにメッセージの着信を通知します。

### 構文

**rcvttty** [ *Command* ]

### 説明

**rcvttty** コマンドは、メールが受信されたことを示すメッセージをユーザーに送信します。**rcvttty** コマンドを始動させるのは、ユーザーではありません。**rcvttty** コマンドは、**/usr/lib/mh/slocal** ファイルが呼び出す **.maildelivery** ファイル内に登録されます。

**rcvttty** コマンドは、1 行のスキャン・リストを端末に送信します。*Command* パラメーターでコマンドを指定すると、**rcvttty** コマンドは、標準入力として着信メッセージを使用して、そのコマンドを実行し、出力を端末に送信します。**rcvttty** コマンドが出力を端末に書き込む場合は、端末の書き込み許可を **all** に設定しなければなりません。

すべての受信メッセージについて **rcvttty** コマンドを実行するには、**.maildelivery** ファイル内で **rcvttty** を指定します。

### フラグ

**-help** コマンド構文、使用可能なスイッチ (トグル)、およびバージョン情報をリストします。

注: MH の場合は、このフラグ名を省略することはできません。

### ファイル

**\$HOME/.maildelivery**

ローカル・メール送達のための命令を提供します。

**\$HOME/.forward**

デフォルトのメッセージ・フィルターを提供します。

**\$HOME/.mh\_profile**

MH ユーザー・プロファイルが入っています。

## 関連情報

**rcvdist** コマンド、 **rcvpack** コマンド、 **rcvstore** コマンド、 **sendmail** コマンド、 **slocal** コマンド。

**.mh\_alias** ファイル・フォーマット。

---

## rdist コマンド

本書では、`usr/bin/rdist` ファイルにある古い AIX `rdist` コマンド、および新規 `rdistd` デーモンに使用される新規 `usr/sbin/rdist` コマンドについて説明しています。

### /usr/bin/rdist コマンド

#### 目的

リモート・ファイル配布クライアント・プログラム。

#### 構文

##### 配布ファイルを使用する

```
rdist [ -n ][ -q ][ -b ][ -D ][ -R ][ -h ][ -i ][ -v ][ -w ][ -y ][ -f FileName ]  
[ -d Argument=Value ][ -m Host ] ... [ Name ] ...
```

##### 引数を小型の配布ファイルとして解釈する

```
rdist [ -n ][ -q ][ -b ][ -D ][ -R ][ -h ][ -i ][ -v ][ -w ][ -y ] -c Name ... [ Login@ ] Host  
[ :Destination]
```

#### 説明

**重要:** `rdist` コマンドを使用して、AIX 4.2 以前のマシンまたは AIX 以外のマシンに 2 ギガバイトより大きいサイズのファイルを送信しないでください。送信すると、未定義の振る舞いという結果となり、まれにデータが失われることがあります。

`rdist` コマンドは、複数のホスト上でファイルの同一のコピーを維持します。`rdist` コマンドは、オーナー、グループ、モード、および可能であればファイルの変更時刻を保存し、さらに実行中のプログラムも更新できます。`rdist` コマンドは、以下のソースから指示を受け取ることができます。

- デフォルトの配布ファイル、すなわちユーザーの `$HOME` ディレクトリー内の `distfile` ファイル
- `-f` フラグで指定する別の配布ファイル
- 配布ファイルの変数定義を追加または上書きするコマンド行の引数
- 小型の配布ファイルとして機能するコマンド行の引数

`-f` フラグを使用しないと、`rdist` コマンドは、ユーザーの `$HOME` ディレクトリー内の `distfile` ファイルを探します。`distfile` ファイルが見つからない場合は、`Distfile` ファイルを探します。

`Name` パラメーターで指定した値は、更新するファイルまたは実行するサブコマンドの名前として読み取られます。コマンド行に `Name` パラメーターの値を指定しないと、`rdist` コマンドは配布ファイルにリストされたすべてのファイルおよびディレクトリーを更新します。`Name` パラメーターに `-` (負符号) を指定すると、`rdist` コマンドは標準入力を使用します。`Name` パラメーターで指定したファイルの名前がサブコマンドの名前と同じ場合は、`rdist` コマンドは `Name` パラメーターをサブコマンドと見なします。

`rdist` コマンドでは、`.rhosts` ファイルを各ホスト上で構成する必要があります。詳細は、*in AIX 5L Version 5.3 Files Reference* `.rhosts` File Format for TCP/IP のセクションを参照してください。

注:

1. **rdist** コマンドがリモート・コンピューターの **/usr/bin/rdist** ディレクトリーに入っていない場合は、**/usr/bin/rdist** ディレクトリーから **rdist** コマンドの実際の位置へのリンクを作成してください。この位置は、通常はディレクトリー **/usr/ucb/rdist** です。
2. 現在、**rdist** コマンドは 7 ビットの ASCII ファイル名しか処理できません。

## フラグ

- b** バイナリー比較を実行して、両者が異なる場合はファイルを更新します。
- c** **rdist** コマンドに残りの引数を小型配布ファイルとして解釈させます。以下の引数が使用可能です。
- Name* 単一の名前、またはブランクで区切られた名前のリストです。値としてファイルまたはサブコマンドが使えます。
- [Login@]Host**  
更新対象となるコンピューターを指定し、オプションとして更新通知先のログイン名を指定します。
- Destination*  
*Name* 引数に単一の名前を指定した場合は、リモート・コンピューター上のファイルを指定します。複数の名前を指定した場合は、ディレクトリーを指定します。
- 注: **-c** フラグを **-f**、**-d**、または **-m** フラグと併用しないでください。
- d Argument=Value** *Argument* 変数の値が *Value* 変数であることを定義します。**-d** フラグは、**distfile** ファイル内の変数定義を定義または上書きします。*Value* 変数は、空文字列、単一の名前、または括弧で囲まれ、タブまたはスペースで区切られた名前のリストとして指定できます。
- D** デバッグ出力をオンにします。
- f FileName** 配布ファイルの名前を指定します。**-f** フラグを使用しなかった場合、デフォルト値は **\$HOME** ディレクトリーの **distfile** ファイルまたは **Distfile** ファイルとなります。
- h** リンク自体ではなく、リンクが指すファイルをコピーします。
- i** 未解決のリンクを無視します。**rdist** コマンドは、転送中のファイルのリンク構造を維持し、すべてのリンクを見つけられなければ、ユーザーに警告します。
- m Host** 更新対象となるコンピューターを制限します。**-m Host** オプションを複数回使用して、更新対象を **distfile** ファイル内にリストされたホストのサブセットに制限することができます。
- n** サブコマンドを実行せずに印刷します。**-n** フラグを使用して **distfile** ファイルをデバッグします。
- q** 抑止モードで操作します。**-q** オプションは、標準出力への修正済みファイルの出力を抑制します。
- R** 無関係のファイルを除去します。ディレクトリーを更新中であれば、リモート・ホスト上に存在してマスター・ディレクトリー内に存在しないファイルが除去されます。**-R** フラグは、ディレクトリーの同一コピーを維持するときに使用します。
- v** ファイルがすべてのホスト上で最新の状態になっているかどうかを検査します。古くなったファイルが表示されます。ただし、**rdist -v** コマンドを実行しても、ファイルは変更されず、メールも送信されません。このフラグを **-b** フラグと一緒に使用すると、このフラグが **-b** フラグをオーバーライドします。
- w** ファイルのパス名全体を宛先ディレクトリー名に追加します。通常、ファイル名を変更してコピー済みファイルのディレクトリー構造を保存するときに **rdist** コマンドが使用するのは、名前の最後のコンポーネントのみです。

~ (ティルド) で始まるファイル名に **-w** フラグを使用すると、ホーム・ディレクトリー以外のすべてが宛先名に追加されます。/ (斜線) または ~ (ティルド) で始まらないファイル名は、宛先ユーザーのホーム・ディレクトリーを残りのファイル名のルート・ディレクトリーとして使用します。

**-y** ファイルの最新のコピーが古くなったファイルで置換されないようにします。一般に、タイム・スタンプとサイズが異なる場合にファイルが更新されます。 **-y** フラグは、**rdist** コマンドがマスター・ファイルよりも新しいファイルを更新しないようにします。

## 配布ファイル (distfile ファイル)

配布ファイルは、コピーするファイル、配布の宛先ホスト、および **rdist** コマンドで配布するファイルの更新時に実行する操作を指定します。通常、**rdist** コマンドは、**\$HOME** ディレクトリーの **distfile** ファイルを使用します。 **-f** フラグを使用すると、別のファイルを指定できます。

## 入力フォーマット

配布ファイルの各エントリーは、以下のいずれかのフォーマットを取ります。

*VariableName = NameList*

配布ファイルのほかのエントリーに使用する変数 (*SourceList*、*DestinationList*、または *SubcommandList*) を定義します。

[*Label*:] *SourceList -> DestinationList SubcommandList*

**rdist** コマンドに対して、*SourceList* 変数で指定したファイルを *DestinationList* 変数で指定したホストに配布するよう指示します。配布ファイルのコマンドは追加機能を実行します。

[*Label*:] *SourceList :: TimeStampFile SubcommandList*

**rdist** コマンドに対して、指定日以降に変更があったファイルを更新するよう指示します。配布ファイルのサブコマンドは追加機能を実行します。 *SourceList* 変数で指定した各ファイルは、タイム・スタンプ・ファイルよりも新しければ更新されます。このフォーマットは、ファイルの復元に役立ちます。

ラベルの指定は任意で、部分的な更新の際にサブコマンドを識別するために使用します。

## 入力

*VariableName*

配布ファイル内で使用する変数を識別します。

*NameList*

ファイルおよびディレクトリー、ホスト、またはサブコマンドのリストを指定します。

*SourceList*

配布用のマスター・コピーとして使用する、**rdist** コマンド用のローカル・ホスト上のファイルおよびディレクトリーを指定します。

*DestinationList*

ホストにファイルのコピーを受信するよう指示します。

*SubcommandList*

実行する配布ファイルのサブコマンドをリストします。

**rdist** コマンドでは、改行文字、タブ、およびブランクをセパレーターとして扱います。展開用の配布ファイル変数は、\$ (ドル記号) で始まり、その後に単一文字または {} (中括弧) で囲まれた名前が続きます。注釈は # (ポンド記号) で始まり、改行文字で終わります。

## ソース・リストおよび宛先リストのフォーマット

配布ファイルのソース・リストおよび宛先リストは、以下に示したフォーマットで示されるように、空白で区切られたゼロまたは複数の名前から構成されます。

[Name1] [Name2] [Name3] ...

**rdist** コマンドは、**csch** コマンドの場合と同様に、ローカル・ホスト上で以下のシェル・メタキャラクターを認識して展開します。

- [ (左大括弧)
- ] (右大括弧)
- { (左中括弧)
- } (右中括弧)
- ( (左括弧)
- ) (右括弧)
- \* (アスタリスク)
- ? (疑問符)

これらの文字が展開されないようにするには、文字の前に \ (バックスラッシュ) を付けます。さらに、**rdist** コマンドは、**csch** コマンドの場合と同様に、~ (ティルド) も展開しますが、ローカル・ホスト上および宛先ホスト上で別々に展開します。

## 配布ファイルのサブコマンド

シェルへの複数のコマンドは、; (セミコロン) で区切る必要があります。コマンドは、更新されるホスト上のユーザーのホーム・ディレクトリ内で実行されます。プログラムを更新した後で専用データベースを再構築するときには、**special** サブコマンドを使用できます。

配布ファイルのサブコマンド・リストには、以下のサブコマンドをゼロ個以上入れることができます。

**install** *Options* [*OptionalDestName*];

古いファイルおよびディレクトリをコピーします。**rdist** コマンドは、各ソース・ファイルまたはソース・ディレクトリを宛先リスト内の各ホストにコピーします。*Options* 変数で指定する使用可能なオプションは、**rdist** コマンド・フラグ **-b**、**-h**、**-i**、**-R**、**-v**、**-w**、および **-y** です。これらのオプションは、*SourceList* 変数で指定したファイルにだけ適用されます。**-R** フラグを使用すると、対応するファイル名がマスター・ホスト上になければ、空でないディレクトリが除去されます。*OptionalDestName* パラメーターはファイル名を変更します。

**notify** *NameList*;

**install** サブコマンドがサブコマンド・リストにない場合、または宛先名を指定していない場合には、ソース・ファイル名が使用されます。リモート・ホスト上に存在しない場合には、パス名の下にディレクトリが作成されます。宛先ホストで使用するログイン名は、宛先名のフォーマットが *login@host* でなければ、ローカル・ホストと同じです。更新ファイル、およびリストされた名前 (*NameList* パラメーター) に生じる可能性のあるエラーのリストをメールします。名前に @ (アットマーク) がなければ、宛先ホストがその名前に追加されます (*name@host*)。

**except** *NameList*;

**rdist** コマンドは、*NameList* 変数で指定したファイルを除き、*SourceList* エントリで指定したすべてのファイルを更新します。



**except\_pat** *NameList*;

**special** *NameList* "String";

**rdist** コマンドが、*NameList* 変数で指定したリストのメンバーと一致する文字列を含むすべてのファイルを更新しないようにします。

*NameList* 変数で指定したファイルの更新またはインストール後にリモート・ホスト上で実行するシェル・コマンド ("String" 変数) を指定します。*NameList* 変数を省略すると、シェル・コマンドは更新またはインストールされたすべてのファイルに対して実行されます。シェル変数 **FILE** は、**rdist** コマンドが "String" 変数を実行する前に、現行ファイル名に設定されます。"String" 値は、" " (二重引用符) で囲む必要があります。配布ファイル内で複数行に渡って指定できます。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0 正常終了。
- >0 エラーが発生したことを示します。

## 例

### フォーマット: **VariableName = NameList** の例

1. 更新するホストのファイルを指定するには、以下のような行を入力します。

```
HOSTS =( matisse root@arpa )
```

ここで、HOSTS 変数は、*matisse* および *root@arpa* として定義されます。**rdist** コマンドは、ホスト *matisse* および *root@arpa* 上のファイルを更新します。この変数を宛先リストとして使用することもできます。

2. *SourceList* エントリーの値として使用する名前を指定するには、以下のような行を入力します。

```
FILES = ( /bin /lib/usr/bin /usr/games  
          /usr/include/{*.h,{stand,sys,vax*,pascal,machine}/*.h}  
          /usr/lib /usr/man/man? /usr/ucb /usr/local/rdist )
```

ここで、FILES の値は、*SourceList* エントリーに使用するファイルとして定義されます。

3. 更新処理から除外するファイルを指定するには、以下のような行を入力します。

```
EXLIB = ( Mail.rc aliases aliases.dir aliases.pag crontab dshrc  
          sendmail.cf sendmail.fc sendmail.hf sendmail.st uucp vfont)
```

ここで、EXLIB の値は、更新処理から除外するファイルのリストとして定義されます。

4. **/usr/src/bin** から **arpa** にすべてのファイルをコピーして、*namelist* 変数を拡張し、それにより *namelist* 変数内にあるファイルと、拡張として **.o** を持つファイル以外の全ファイルがコピーされるようにするには、以下のような行を入力します。

```
/usr/src/bin ->arpa  
except_pat(\e\e.o\e ${<namelist> /SCCS\e ${<namelist>}
```

または

```
/usr/src/bin ->arpa  
except_pat(\\.o\e ${<namelist> /SCCS\e ${<namelist>}
```

5. **.o** という拡張を持つファイル以外のすべてのファイルを **/usr/src/bin** から **arpa** にコピーするには、以下のような行を入力します。



```
/usr/src/bin ->arpa
except_pat(\\.o$ /SCCS$)
```

## フォーマット: [label:] SourceList - DestinationList SubcommandList の例

1. ファイルのソース・リストをホストの宛先リストにコピーするには、以下のような行を入力します。

```
/${FILES} ->${HOSTS}
install -R
except /usr/lib/${EXLIB} ;
except /usr/games/lib ;
special /usr/sbin/sendmail "/usr/sbin/sendmail.bz" ;
```

上記の行で、[Label:] エントリーの指定は、任意であるため、ここには表示してありません。\$ (ドル記号) および {} (中括弧) によって、ファイル名 FILES、HOSTS、および EXLIB が前の例で指定したリストに展開されます。この例の残りの部分は、サブコマンド・リストを構成します。

2. [Label:] エントリーを使用するには、以下のような行を入力します。

```
srcsL:
/usr/src/bin -> arpa
except_pat (\\.e.o$ /SCCS$ ) ;
```

ラベルは srcsL: で、更新の際にこのエントリーの識別に使用できます。/usr/src/bin ファイルはコピー元のソースで、ホスト arpa はコピーの宛先です。3 行目は、サブコマンド・リストからのサブコマンドです。

3. タイム・スタンプ・ファイルを使用するには、以下のような行を入力します。

```
/${FILES} :: stamp.cory
notify root@cory
```

\$ (ドル記号) および {} (中括弧) によって FILES で指定した名前が、例で指定したリストに展開されます。タイム・スタンプ・ファイルは stamp.cory です。最終行は、サブコマンド・リストからのサブコマンドです。

## ファイル

|                        |                                          |
|------------------------|------------------------------------------|
| <b>/usr/bin/rdist</b>  | <b>rdist</b> コマンドが入っています。                |
| <b>\$HOME/distfile</b> | <b>rdist</b> コマンドが読み取るサブコマンドのリストが入っています。 |
| <b>/tmp/rdist</b>      | 更新リストが入っています。これは一時ファイルです。                |

## 関連情報

**cs** コマンド、 **ksh** コマンド、 **sh** コマンド。

**.rhosts** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ の通信およびネットワークのセクション。

## /usr/sbin/rdist コマンド

本書では、**usr/bin/rdist** ファイルにある古い AIX **rdist** コマンド、および新規 **rdistd** デーモンに使用される新規 **usr/sbin/rdist** コマンドについて説明しています。

## 目的

ファイルをリモート側で配布するクライアント・プログラム

## 構文

### 配布ファイルを使用する

```
/usr/sbin/rdist [ -Fn ] [ -A num ] [ -a num ] [ -d var=value ] [ -l < local logopts > ] [ -L < remote logopts > ] [ -f distfile ] [ -M maxproc -m host ] [ -o distops ] [ -t timeout ] [ -p < rdist-path > ] [ -P < transport-path > ] [ name... ]
```

### 引数を小型の配布ファイルとして解釈する

```
/usr/sbin/rdist -Fn -c name ... [ login@ ] host [ :dest ]
```

### 古い rdist をサーバーとして起動する

```
/usr/sbin/rdist -Server
```

### バージョン情報

```
/usr/sbin/rdist -V
```

## 説明

**rdist** は、複数のホストでファイルの同一のコピーを保守するプログラムです。これは、オーナー、グループ、モード、および可能であればファイルの変更時刻を保存し、さらに実行中のプログラムを更新できます。**rdist** コマンドは、以下のソースから指示を受け取ることができます。

- 現行ディレクトリー内の配布ファイル **distfile**。
- **distfile** が - と指定されている場合は標準入力。
- **-f** フラグを使用していない場合は、**rdist** は **distfile** および **Distfile** という名前のファイルを探します。
- **-c** フラグを使用した場合は、末尾の引数は小さな **distfile** として解釈されます。等しい **distfile** は以下のとおりです。

```
( filename ... ) -> [user@]host
install          [dest name] ;
```

**name** 引数が指定されていない場合、**rdist** は、**distfile** にリストされたすべてのファイルおよびディレクトリーを更新します。これ以外の場合、引数は更新するファイルの名前、または実行するコマンドのラベルであると解釈されます。ラベルとファイル名が矛盾する場合は、ラベルであると想定されます。これらを一緒に使用し、特定のコマンドを使用して特定のファイルを更新することができます。

**-Server** オプションは、**rdist** の古いバージョン (**rdist** をサーバー・モードにするため、このオプションを使用) のための、後ろ向きの互換性を提供します。**-Server** コマンド行オプションを使用して **rdist** を開始すると、**rdist** の古いバージョンを実行します。このオプションは、古い **rdist** が **/usr/bin/rdist** にある場合のみ働きます。

**rdist** は任意のトランスポート・プログラムを使用して、それぞれのターゲット・ホストにアクセスします。トランスポート・プログラムは、**-P** フラグを使用して、コマンド行に指定することができます。**-P** フラグを使用しないと、**rsh** がトランスポート・プログラムになります。**rsh** メソッドを使用し、ターゲット・ホストが文字列 **localhost** であり、リモート・ユーザー名がローカル・ユーザー名と同じ場合、**rdist** は次のコマンドの実行を試みます。

```
/bin/sh -c rdistd -S
```

これ以外の場合、**rdist** は次のコマンドを実行します。

```
rsh host -l remuser rdistd -S
```

上の例で、*host* パラメーターはターゲット・ホストの名前、*remuser* は接続を行うユーザーの名前、および **rdistd** はターゲット・ホストの **rdist** サーバー・コマンドです。

トランスポート・プログラムは、**rsh** の場合、上記の構文と互換性がある必要があります。互換性がない場合には、このコマンド行構文を理解するシェル・スクリプト内に、トランスポート・プログラムを含める必要があります。

それぞれのターゲット・ホスト上で、**rdist** は以下のコマンドを実行します。

```
rdistd -S
```

または

```
<rdistd path> -S
```

上の例は、**-p** フラグを指定しました。**-p** フラグを指定しない、または `<rdistd path>` が単純なファイル名の場合、**rdistd** または `<rdistd path>` は、リモート (ターゲット) ホストで **rdist** を実行するユーザーの **PATH** 内のどこかになければなりません。

**rdist** コマンドは次の環境変数を使用します。

**TMPDIR**

使用する一時ディレクトリーの名前。デフォルトは **/tmp** です。

## フラグ

|                       |                                                                                                                                                                         |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-A num</b>         | ファイルシステムに最小限のフリー・ファイル (inodes) が存在する場合のみ、ファイルを更新またはインストールします。                                                                                                           |
| <b>-a num</b>         | ファイルシステムに最小限のフリー・スペースが存在する場合のみ、ファイルを更新またはインストールします。                                                                                                                     |
| <b>-d var = value</b> | <i>value</i> を変数 <i>var</i> に割り当てます。このオプションは、 <b>distfile</b> 内の変数定義を定義または変更するために使用します。 <i>Value</i> は、空の文字列、単一の名前、または括弧で囲まれ、タブまたはスペース (またはその両方) で区切られた名前リストにすることができます。 |
| <b>-F</b>             | 子プロセスを <code>fork</code> せずに、すべてのクライアントを順番に更新します。                                                                                                                       |
| <b>-f distfile</b>    | <b>distfile</b> を配布ファイルとして使用します。 <b>distfile</b> が <code>-</code> と指定されている場合は、標準入力から読み取ります。                                                                             |
| <b>-l logopts</b>     | ローカルのロギング・オプションを設定します。 <i>logopts</i> の構文については、 <b>メッセージ・ログ</b> のセクションを参照してください。                                                                                        |
| <b>-L logopts</b>     | リモートのロギング・オプションを設定します。 <i>logopts</i> は、リモート・サーバーに値が渡される ( <b>rdistd</b> ) こと以外は、ローカル・ロギングと同じです。 <i>logopts</i> の構文については、 <b>メッセージ・ログ</b> のセクションを参照してください。              |
| <b>-M num</b>         | 同時に実行する子 <b>rdist</b> プロセスの最大数を <i>num</i> に制限します。デフォルト値は 4 です。                                                                                                         |
| <b>-m machine</b>     | ファイルの更新を指定したマシンに制限します。複数の <b>-m</b> 引数を指定し、更新を、 <b>distfile</b> にリストされたホストのサブセットに制限することができます。                                                                           |
| <b>-n</b>             | コマンドを表示しますが、実行はしません。 <b>-n</b> フラグを使用して <b>distfile</b> をデバッグします。                                                                                                       |

-o *distopts*

*dist* オプションを使用可能にします。 *distopts* は、以下にリストされたオプションをコンマで区切ったリストです。 *distopts* に有効な値は以下のものです。

**chknfs** ターゲット・ファイルシステムが NFS の場合、ファイルを検査または更新しません。

**chkreadonly**

ターゲット・ホストのファイルが読み取り専用ファイルシステムに存在する場合、ファイルの検査も更新も行われません。

**chksym** リモート・ホスト上のターゲットがシンボリック・リンクであるが、マスター・ホスト上にない場合、リモート・ターゲットはシンボリック・リンクのままです。

**compare** ファイルにバイナリー比較を実行して、両者が異なる場合はファイルを更新します。

**follow** リンク自体ではなく、シンボリック・リンクが指すファイルをコピーします。

**ignlnks** 解決できないリンクを無視します。この場合、**rdist** は通常、未解決のリンクについてユーザーに警告を出します。

**nochkowner**

ファイルが既に存在する場合、ユーザーの所有権を検査しません。ファイル所有権は、ファイルの更新時にも設定されます。

**nochkgroup**

ファイルが既に存在する場合、グループの所有権を検査しません。ファイル所有権は、ファイルの更新時にも設定されます。

**nochkmode**

ファイルおよびディレクトリー許可モードの検査をしません。許可モードは、ファイルの更新時にも設定されます。

**nodescend**

ディレクトリーに再帰的に下降することはしません。ディレクトリーの存在、所有権、およびモードだけが検査されます。

**noexec** **a.out** フォーマットの実行可能ファイルの検査または更新を行いません。

**numchkgroup**

グループ名ではなく数字のグループ ID (gid) を使用して、グループ所有権を検査します。

**numchkowner**

ユーザー名ではなく数字のユーザー ID (uid) を使用して、ユーザー所有権を検査します。

**quiet** 標準出力上で変更されるファイルの印刷を抑止します。

**remove** リモート・ホスト上に存在するディレクトリー内のファイルで、ローカル・ホストのマスター・ディレクトリーに存在しないファイルをすべて除去します。

**savetargets**

更新されるファイルを除去するのではなく、保管します。更新されるターゲット・ファイルは、まず最初に **filename** から **filename.OLD** に名前変更されます。

**sparse** スパース・ファイルの検査を使用可能にします。このオプションはある程度の追加の処理オーバーヘッドを必要とするので、スパース・ファイルを含んでいる可能性のあるターゲットにのみ使用可能にすべきです。

(*dist* オプションの続き):

**verify** すべてのホストのすべての期限切れのファイルを表示しますが、ファイルは変更されず、メールも送信されません。

**whole** 宛先ディレクトリー名にファイル名全体を付加します。通常、ファイルの名前変更には、名前の最後のコンポーネントのみが使用されます。これにより、コピーされるファイルのディレクトリー構造が維持され、ディレクトリー構造が平坦になりません。例えば、**/path/dir1/f1** および **/path/dir2/f2** のファイルのリストを **/tmp/dir** に **rdist** すると、**/tmp/dir/dir1/f1** と **/tmp/dir/dir2/f2** ではなく、ファイル **/tmp/dir/path/dir1/f1** と **/tmp/dir/path/dir2/f2** が作成されます。

**younger** ファイルは通常、**mtime** と **size** が一致しない場合に更新されます。このオプションは、マスター・コピーよりも若いファイルを **rdist** が更新しないようにします。これは、他のホスト上の、より新しいコピーが置き換えられるのを防ぐために使用することができます。マスター・コピーよりも新しいファイルについては、警告メッセージが印刷されます。

-o *distopts*

-p *<rdist-path>*

ターゲット・ホスト上の指定されたパス内で **rdistd** サーバーを探します。

**-P** <*rdist-path*> *transport-path* で指定されたトランスポート・プログラムを使用します。 *transport-path* は、可能なバス名をコロンで区切ったリストにすることができます。この場合、存在するバスの最初のコンポーネントが使用されます。

**-t** *timeout* リモートの **rdist** サーバーからの応答を待つ *timeout* 期間 (秒単位) を設定します。デフォルトは 900 秒です。

**-V** バージョン情報を印刷して、終了します。

## メッセージ・ログ

**rdist** コマンドは、メッセージ機能のセットを提供し、各機能には、その機能に送信するメッセージのタイプを示す、メッセージ・タイプのリストが含まれます。ローカル・クライアント (**rdist**) とリモート・サーバー (**rdistd**) はそれぞれ、どの機能にどのタイプのメッセージをログ記録するかについて、別のコピーを保守します。

**-l** *logopts* フラグは、クライアント上でローカルに使用するロギング・オプションを指定します。 **-L** *logopts* フラグは、リモート **rdistd** サーバーに渡すロギング・オプションを指定します。

*logopts* のフォーマットは以下のとおりです。

facility=types:facility= types...

有効な機能名は以下のとおりです。

### stdout

メッセージを標準出力に出力。

**file** メッセージはファイルに送信されます。ファイル名は **file** = *filename* = *types* のフォーマットで指定します。

### syslog

メッセージは **syslogd** 機能に送信されます。

**notify** メッセージは内部 **rdistnotify** 機能に送信されます。この機能は、**distfile** 内の **notify** キーワードと組み合わせて使用し、どのメッセージを **notify** アドレスにメールするかを指定します。

*types* は、メッセージ・タイプをコンマで区切って指定したリストにする必要があります。指定する各メッセージ・タイプは、そのメッセージ・レベルを使用可能にします。昇順方式を使用する **syslog** システム機能とは、この点が異なります。有効なタイプを以下に示します。

### change

変更するものについて、メッセージをログに記録します。

**info** 一般情報をログに記録します。

**notice** 変更するものについて、一般情報のメッセージをログに記録します。変更するものには、**distfile** には明示的に指定されていないが、特定のターゲットにインストールするために必要なディレクトリーを作成した、といったものも含まれます。

**error** 致命的ではない通常のエラーのメッセージをログに記録します。

**fatal** 致命的エラーのメッセージをログに記録します。

### warning

**error** タイプのメッセージほど重大ではないエラーについての警告をログに記録します。

### verbose

通常よりも詳細であるが、デバッグ・レベルよりは少ないメッセージをログに記録します。

**debug** デバッグ情報をログに記録します。

**all** デバッグ・メッセージを除くすべてをログに記録します。

## 配布ファイル

配布ファイルは、コピーするファイル、配布の宛先ホスト、および **rdist** コマンドで配布するファイルの更新時に実行する操作を指定します。

## 入力フォーマット

配布ファイルの各エントリは、以下のいずれかのフォーマットを取ります。

VariableName = NameList

配布ファイルの他のエントリ (*SourceList*、*DestinationList*、または *SubcommandList*) に使用される変数を定義します。

[Label:] SourceList -> DestinationList SubcommandList

**rdist** コマンドに対して、*SourceList* 変数で指定したファイルを *DestinationList* 変数で指定したホストに配布するよう指示します。配布ファイルのコマンドは追加機能を実行します。

[Label:] SourceList :: TimeStampFile SubcommandList

**rdist** コマンドに対して、指定日以降に変更があったファイルを更新するよう指示します。配布ファイルのサブコマンドは追加機能を実行します。 *SourceList* 変数で指定した各ファイルは、タイム・スタンプ・ファイルよりも新しければ更新されます。

ラベルはオプションです。ラベルは、部分的更新用のコマンドを識別するために使用されます。

## 入力

*VariableName*

配布ファイル内で使用する変数を識別します。

*NameList*

ファイルおよびディレクトリ、ホスト、またはサブコマンドのリストを指定します。

*SourceList*

配布用のマスター・コピーとして使用する、**rdist** コマンド用のローカル・ホスト上のファイルおよびディレクトリを指定します。

*DestinationList*

ホストにファイルのコピーを受信するよう指示します。

*SubcommandList*

実行する配布ファイルのサブコマンドをリストします。

**rdist** コマンドでは、改行文字、タブ、およびブランクをセパレーターとして扱います。展開用の配布ファイル変数は、\$ (ドル記号) で始まり、その後単一文字または中括弧で囲まれた名前が続きます。コメントはポンド記号 (#) で始まり、改行文字で終わります。

## ソース・リストおよび宛先リストのフォーマット

配布ファイルのソース・リストおよび宛先リストは、以下に示したフォーマットで示されるように、ブランクで区切られたゼロまたは複数の名前から構成されます。

[Name1] [Name2] [Name3] ...

**rdist** コマンドは、**csH** コマンドの場合と同様に、ローカル・ホスト上で以下のシェル・メタキャラクターを認識して展開します。

- [ 左大括弧
- ] 右大括弧
- { 左中括弧
- } 右中括弧



- ( 左括弧
- ) 右括弧
- \* アスタリスク
- ? 疑問符

これらの文字が展開されないようにするには、文字の前に \ (バックスラッシュ) を付けます。 **rdist** コマンドは、 **csch** コマンドと同様にティルドを展開しますが、ローカル・ホストと宛先ホストで別々に展開します。ティルドで始まるファイル名に **-o whole** オプションを使用すると、ホーム・ディレクトリー以外のすべてが宛先名に付加されます。 / (斜線) または ~ (ティルド) で始まらないファイル名は、宛先ユーザーのホーム・ディレクトリーを残りのファイル名のルート・ディレクトリーとして使用します。

## 配布ファイルのサブコマンド

シェルへの複数のコマンドは、 ; (セミコロン) で区切る必要があります。コマンドは、更新されるホスト上のユーザーのホーム・ディレクトリー内で実行されます。 プログラムを更新した後で専用データベースを再構築するときには、 **special** サブコマンドを使用できます。

配布ファイルのサブコマンド・リストには、以下のサブコマンドをゼロ個以上入れることができます。

**install** Options[OptionalDestName];

古いファイルおよびディレクトリーをコピーします。 **rdist** コマンドは、各ソース・ファイルまたはソース・ディレクトリーを宛先リスト内の各ホストにコピーします。 *Options* 変数で指定する使用可能なオプションは、 **rdist** コマンド・フラグ **-b**、 **-h**、 **-i**、 **-R**、 **-v**、 **-w**、 および **-y** です。これらのオプションは、 *SourceList* 変数で指定したファイルにだけ適用されます。 **-R** フラグを使用すると、対応するファイル名がマスター・ホスト上になければ、空でないディレクトリーが除去されます。 *OptionalDestName* パラメーターはファイル名を変更します。

**notify** NameList;

**install** サブコマンドがサブコマンド・リストにない場合、または宛先名を指定していない場合には、ソース・ファイル名が使用されます。リモート・ホスト上に存在しない場合には、パス名の下にディレクトリーが作成されます。宛先ホストで使用するログイン名は、宛先名のフォーマットが `login@host` でなければ、ローカル・ホストと同じです。

更新ファイル、およびリストされた名前 (*NameList* パラメーター) に生じる可能性のあるエラーのリストをメールします。名前に @ (アットマーク) がなければ、宛先ホストがその名前に追加されます (`name@host`)。

**except** NameList;

**rdist** コマンドは、 *NameList* 変数で指定したファイルを除き、 *SourceList* エントリーで指定したすべてのファイルを更新します。

**except\_pat** NameList;

**rdist** コマンドが、 *NameList* 変数で指定したリストのメンバーと一致する文字列を含むすべてのファイルを更新しないようにします。



special NameList "String";

*NameList* 変数で指定したファイルの更新またはインストール後にリモート・ホスト上で実行するシェル・コマンド ("String" 変数) を指定します。*NameList* 変数を省略すると、シェル・コマンドは更新またはインストールされたすべてのファイルに対して実行されます。シェル変数 *FILE* は、**rdist** コマンドが "String" 変数を実行する前に現行ファイル名にセットされ、変数 *REMFIL*E には、たった今更新されたリモート・ファイルの完全パス名が含まれ、変数 *BASEFILE* には、たった今更新されたリモート・ファイルの基本名が含まれます。"String" 値は二重引用符で囲む必要があります、配布ファイル内で複数行にまたがることができます。

cmdspecial NameList "String";

**cmdspecial** コマンドは **special** と似ていますが、各ファイルが更新されるたびにではなく、全体のコマンドが完了した後にのみ実行される点が異なります。シェル変数 *FILES* にはファイルのリストが含まれます。*FILES* シェル変数内の各ファイル名はコロンで区切ります。

NFS は、ホスト名が正符号 (+) で終わる場合は使用不可になります。これは、**-o chknfs** オプションをこの 1 つのホストにだけ使用不可にすることと同等です。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0 正常終了。
- >0 エラーが発生しました。

## 例

1. 更新するホストのファイルを指定するには、以下のような行を入力します。

```
HOSTS =( matisse root@arpa )
```

上の例で、*HOSTS* 変数は *matisse* および *root@arpa* として定義されます。**rdist** コマンドは、ホスト *matisse* および *root@arpa* 上のファイルを更新します。この変数を宛先リストとして使用することもできます。

2. *SourceList* エントリーの値として使用する名前を指定するには、以下のような行を入力します。

```
FILES = ( /bin /lib/usr/bin /usr/games  
          /usr/include/{*.h,{stand,sys,vax*,pascal,machine}/*.h}  
          /usr/lib /usr/man/man? /usr/ucb /usr/local/rdist )
```

上の例で、*FILES* 値は *SourceList* エントリーに使用するファイルとして定義されます。

3. 更新処理から除外するファイルを指定するには、以下のような行を入力します。

```
EXLIB = ( Mail.rc aliases aliases.dir aliases.pag crontab dshrc  
          sendmail.cf sendmail.fc sendmail.hf sendmail.st uucp vfont)
```

上の例で、*EXLIB* 値は、更新処理から除外するファイルのリストを定義します。

4. */usr/src/bin* から *arpa* にすべてのファイルをコピーして、*namelist* 変数を拡張し、それにより *namelist* 変数内にあるファイルと、拡張として *.o* を持つファイル以外の全ファイルがコピーされるようにするには、以下のような行を入力します。

```
/usr/src/bin ->arpa
except_pat(\e\e.o\e ${<namelist> /SCCS\e ${<namelist>}
```

または

```
/usr/src/bin ->arpa
except_pat(\\.o\e ${<namelist> /SCCS\e ${<namelist>}
```

5. .o という拡張を持つファイル以外のすべてのファイルを /usr/src/bin から arpa にコピーするには、以下のような行を入力します。

```
/usr/src/bin ->arpa
except_pat(\\.o\${ /SCCS\${
```

## フォーマット: [label:] SourceList - DestinationList SubcommandList の例

1. ファイルのソース・リストをホストの宛先リストにコピーするには、以下のような行を入力します。

```
${FILES} ->${HOSTS}
install -R
except /usr/lib/${EXLIB} ;
except /usr/games/lib ;
special /usr/sbin/sendmail "/usr/sbin/sendmail.bz" ;
```

上記の行で、[Label:] エントリーの指定は、任意であるため、ここには表示してありません。\$ (ドル記号) および {} (中括弧) によって、ファイル名 FILES、HOSTS、および EXLIB が前の例で指定したリストに展開されます。この例の残りの部分は、サブコマンド・リストを構成します。

2. [Label:] エントリーを使用するには、以下のような行を入力します。

```
srcsL:
/usr/src/bin -> arpa
except_pat (\e\e.o\e$ /SCCS\e$ ) ;
```

ラベルは srcsL: で、更新の際にこのエントリーの識別に使用できます。**/usr/src/bin** ファイルはコピー元のソースで、ホスト arpa はコピーの宛先です。3 行目は、サブコマンド・リストからのサブコマンドです。

3. タイム・スタンプ・ファイルを使用するには、以下のような行を入力します。

```
${FILES} :: stamp.cory
notify root@cory
```

\$ (ドル記号) および {} (中括弧) によって、FILES で指定した名前が指定したリストに展開されます。タイム・スタンプ・ファイルは **stamp.cory** です。最終行は、サブコマンド・リストからのサブコマンドです。

## ファイル

|                         |                                         |
|-------------------------|-----------------------------------------|
| <b>/usr/sbin/rdist</b>  | バージョン 6.1.5 の <b>rdist</b> コマンドが入っています。 |
| <b>distfile</b>         | 入力コマンドが入っています。                          |
| <b>\$ TMPDIR/rdist*</b> | 更新リスト用の一時ファイルです。                        |

## 関連情報

**cs**h コマンド、**rcmd()** システム・コール、**rdistd** コマンド、**rsh** コマンド、**sh** コマンド、**stat()** システム・コール。

---

## rdistd コマンド

### 目的

ファイルをリモート側で配布するサーバー・プログラム。

### 構文

`rdistd -S`

`rdistd -V`

### 説明

`rdistd` は、`rdist` コマンドのサーバー・プログラムです。これは通常、`rsh` を使用して `rdist` により実行されます。

`rdistd` は、通常ユーザーの `PATH` 環境変数内にあるので、`-S` フラグは、`rdistd` が誤って開始されないようにします。

### フラグ

`-V` バージョン情報を印刷し、終了します。

### 終了状況

このコマンドは、以下の終了値を戻します。

0 正常終了。

>0 エラーが発生しました。

### ファイル

`/usr/sbin/rdistd`

`/usr/bin/rdistd`

`rdistd` サーバーが入っています。

`/usr/sbin/rdistd` へのシンボリック・リンク。

### 関連情報

675 ページの『`rdist` コマンド』 および 892 ページの『`rsh` または `remsh` コマンド』。

---

## rdump コマンド

### 目的

リモート・マシンのデバイス上にファイルのバックアップを取ります。

注: このコマンドを実行するユーザーには、`root` 権限が必要です。

### 構文

`rdump [ -b Blocks ] [ -B ] [ -c ] [ -d Density ] [ -L Length ] [ -s Size ] [ -u ] [ -w ] [ -W ] [ -Level ] -f Machine:Device [ FileSystem | DeviceName ]`

## 説明

**rdump** コマンドは、ローカル・コンピューターのファイルシステムを *i* ノードによってリモート・コンピューターにコピーします。ファイルは、**backup** コマンド・フォーマットを使用して、リモート・コンピューターのデバイスにコピーされます。デバイスへのアクセスには、リモート・マシン上のリモート・サーバーを使います。**rdump** コマンドを実行するには、**root** 権限を持っていなければなりません。また、宛先のリモート・コンピューターの **/rhosts** ファイル内に、**rdump** コマンドを実行するローカル・コンピューターを定義しておかなければなりません。

ファイルシステムをバックアップするには、バックアップしたいファイルを示す **-Level** パラメーターおよび **FileSystem** パラメーターを指定します。**-Level** パラメーターを使用すると、システム上のすべてのファイルをバックアップするか (フル・バックアップ)、特定のフル・バックアップ以降に修正されたファイルのみをバックアップ (増分バックアップ) することができます。指定可能なレベルは 0 から 9 までです。レベルの指定がないと、デフォルト・レベルの 9 が使われます。レベル 0 のバックアップには、ファイルシステム上のすべてのファイルが含まれます。レベル *n* のバックアップには、最後のレベル *n* - 1 (*n* マイナス 1) のバックアップ以降に修正されたすべてのファイルが含まれます。レベルを **-u** フラグと一緒に使用すると、ファイルシステムごと増分バックアップの階層を維持することができます。

### 注:

1. 増分バックアップ (**-Level** パラメーター) を実行して、増分バックアップごとに最終日、時刻、およびレベルに関する情報が **/etc/dumpdates** ファイルに書き込まれるようにするには、**-u** フラグを使用します。
2. リモート・コンピューター上の **rmt** コマンドが **/usr/sbin/rmt** に入っていない場合は、リモート・コンピューター上で **/usr/sbin/rmt** から実際の位置 (通常は **/etc/rmt**) へのリンクを作成する必要があります。

## フラグ

- b Blocks** 一回の出力操作で書き込むブロック数を指定します。**Blocks** 変数を指定しなければ、**rdump** コマンドは選択された物理デバイス用の適切なデフォルト値を使用します。**Blocks** 変数の値が大きくなるほど、テープ・デバイスへの物理的な転送は大きくなります。
- B** エラーの発生時に、ユーザーに照会せずにコマンドを終了させます。**-B** フラグを指定した場合、**rdump** コマンドはゼロ以外の値を戻します。
- c** テープが 9 トラック・フォーマットではなくカートリッジ・フォーマットであることを指定します。
- d Density** テープ密度を 1 インチ当たりのビット数 (bpi) 単位で指定します。この値は、ボリューム当たりのテープ使用量を計算するときに使われます。**Density** 変数の値を指定しなければ、デフォルト密度は 1600 bpi となります。テープ密度を指定せずに **-c** フラグを使用すると、デフォルト密度は 8000 bpi になります。
- fMachine:Device** **Machine** 変数をリモート・コンピューターのホスト名として指定します。指定したデバイスに出力を送信するには、**Device** 変数をファイル名 (**/dev/rmt0** ファイルなど) として指定します。**Device** 変数には、テープ・デバイスのみを指定してください。
- L Length** テープの長さをバイト単位で指定します。このフラグは、**-c**、**-d**、および **-s** フラグを上書きします。それぞれブロック (512 バイト)、キロ (1024 バイト)、メガ (1024 キロバイト)、またはギガ (1024 メガバイト) を示す接尾部 **b**、**k**、**m**、または **g** でサイズを指定できます。2 ギガバイトのテープの長さを表すには、**-L 2g** と入力します。このフラグは、AIX 4.2 以上のみ適用されます。

|                       |                                                                                                                                                                                                           |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-s</b> <i>Size</i> | <i>Size</i> 変数を使用して、テープのサイズをフィート単位で指定します。テープ・サイズを指定しなければ、デフォルト・サイズは 2300 フィートとなります。テープ・サイズを指定せずに <b>-c</b> フラグを使用すると、デフォルト・サイズは 1700 フィートとなります。テープ・ドライブが指定されたサイズに達すると、 <b>rdump</b> コマンドはテープが交換されるまで待ちます。 |
| <b>-u</b>             | <b>/etc/dumpdates</b> ファイル内でリモート・バックアップの時刻、日付、およびレベルを更新します。このファイルには、増分バックアップの維持に必要な情報が入っています。                                                                                                             |
| <b>-w</b>             | 現在使用不可。                                                                                                                                                                                                   |
| <b>-W</b>             | <b>/etc/dumpdates</b> ファイルで検出されたファイルシステムを表示します。                                                                                                                                                           |
| <b>-Level</b>         | リモート・バックアップ・レベル (0 から 9 まで) を指定します。 <i>Level</i> 変数のデフォルト値は 9 です。                                                                                                                                         |
| <b>-?</b>             | 使用方法メッセージを表示します。                                                                                                                                                                                          |

## パラメーター

|                   |                                                                                                                                                         |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>DeviceName</i> | 物理デバイス名 (ブロック名またはロー名) を指定します。                                                                                                                           |
| <i>FileSystem</i> | ファイルシステムが通常マウントされるディレクトリの名前を指定します。 <b>rdump</b> コマンドは、物理デバイス名として <b>/etc/filesystems</b> を読み込みます。ファイルシステムを指定しないと、デフォルトは、ルート ( <i>/</i> ) ファイルシステムになります。 |

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0 コマンドが正常に終了したことを示します。
- >0 エラーが発生したことを示します。

## 例

- ローカル・コンピューター上の **/usr** ディレクトリー内のファイルをリモート・コンピューターにバックアップするには、以下のように入力します。

```
rdump -u -0 -fcanine:/dev/rmt0 /usr
```

**-u** フラグは、システムに対して、**/etc/dumpdates** ファイル内の現行バックアップ・レベル・コードを更新するように指示します。**-Level** フラグをバックアップ・レベル 0 に設定すると、**/usr** ディレクトリー内のすべてのファイルをバックアップするように指定したことになります。リモート・コンピューターの ID は **canine** で、デバイスは **/dev/rmt0** デバイスです。

- 8mm、2.3 ギガバイトのテープを使用して、ローカル・コンピューター上の **/usr** ディレクトリー内のファイルをリモート・コンピューターにバックアップするには、以下のように入力します。

```
rdump -fcanine:/dev/rmt0 -L 2200m /usr
```

**注:** ここでは、2.3 ギガバイトの代わりに 2.2 ギガバイトを使用して、テープの実際の終わりをヒットしないようにします。

- 0.25 インチ・テープを使用してローカル・コンピューター上の **/usr** ディレクトリー内のファイルをリモート・コンピューターにバックアップするには、以下のように入力します。

```
rdump -fcanine:/dev/rmt0 -c /usr
```

-c フラグを使用すると、**rdump** コマンドはデフォルトとして 0.25 インチ・テープに合ったサイズ値と密度値を使用します。

## ファイル

|                         |                                           |
|-------------------------|-------------------------------------------|
| <b>/etc/dumpdates</b>   | リモート・ダンプをとった最新の日付のログが入っています。              |
| <b>/etc/filesystems</b> | ファイルシステムの情報が入っています。                       |
| <b>/dev/rhd4</b>        | デフォルトのファイルシステム (ルート) が配置されているデバイスが入っています。 |
| <b>/usr/sbin/rdump</b>  | <b>rdump</b> コマンドが入っています。                 |

## 関連情報

**find** コマンド、**dd** コマンド、**backup** コマンド、**restore** コマンド、**rrestore** コマンド。

**dumpdates** ファイル、**filesystems** ファイル、**rmt** スペシャル・ファイル。

オペレーティング・システムおよびデバイスのマネージのバックアップ方式、ディレクトリー、ファイルのセクション。

オペレーティング・システムおよびデバイスのマネージのファイルシステムのセクション。

---

## read コマンド

### 目的

標準入力から 1 行読み取ります。

### 構文

```
read [ -p ] [ -r ] [ -s ] [ -u[ n ] ] [ VariableName?Prompt ]
```

```
[ VariableName ... ]
```

### 説明

**read** コマンドは、標準入力から一行読み取り、IFS (内部フィールド・セパレーター) 変数の文字をセパレーターとして使用して、入力行内の各フィールドの値をシェル変数に割り当てます。 *VariableName* パラメーターは、入力行から 1 つのフィールドの値を取り込んだシェル変数の名前を指定します。 *VariableName* パラメーターで指定した最初のシェル変数には、最初のフィールドの値が代入され、 *VariableName* パラメーターで指定した 2 番目のシェル変数には 2 番目のフィールドの値が代入されるというように、最後のフィールドに達するまで、値が代入されていきます。標準入力行のフィールド数が *VariableName* パラメーターで指定した対応するシェル変数よりも多い場合は、最後に指定したシェル変数に残りの全フィールドの値が代入されます。フィールド数がシェル変数よりも少ない場合は、残りのシェル変数は空文字列に設定されます。

注: *VariableName* パラメーターを省略すると、**REPLY** 変数をデフォルトの変数名として使用されます。

**read** コマンドによってシェル変数を設定すると、現在のシェル実行環境はその影響を受けます。

## フラグ

- p** l& (パイプ接続、アンパーサンド) を使用して、Korn シェルが実行したプロセスの出力から入力を読み取ります。  
注: **-p** フラグにファイル終了文字を指定すると、別のプロセスを作成するためにこのプロセスのクリーンアップが実行されます。
- r** read コマンドが \ (バックスラッシュ) 文字を制御文字としてではなく、入力行の一部として処理するように指定します。
- s** 入力をコマンドとして Korn シェルのヒストリー・ファイルに保存します。
- u [ n ]** 1 桁のファイル・ディスクリプター数値 *n* から入力データを読み取ります。このファイル・ディスクリプターは、ksh exec 組み込みコマンドでオープンすることができます。*n* のデフォルト値は 0 で、これはキーボードを示します。値 2 は標準エラーを示します。

## パラメーター

- VariableName?Prompt* 変数の名前と使用されるプロンプトを指定します。対話式 Korn シェルはプロンプトを標準エラーに書き込んで、入力を実行します。 *Prompt* に複数のワードが含まれる場合、必ず各ワードを単一引用符または二重引用符で囲んでください。
- VariableName...* 1 つまたはそれ以上の変数名をホワイト・スペースで区切って指定します。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0 正常終了。
- >0 ファイル終わり文字が検出されたか、またはエラーが発生しました。

## 例

1. 次のスクリプトは、ファイルの各行の先頭のフィールドを行末に移動して出力します。

```
while read -r xx yy
do
    print printf "%s %s/n" $yy $xx
done < InputFile
```

2. 読み取った行を複数フィールドに分割し、"Please enter: " をプロンプトとして使用するには、次のように入力します。

```
read word1?"Please enter: " word2
```

システムは次のように表示します。

```
Please enter:
You enter:
hello world
```

変数 *word1* の値が "hello" で変数 *word2* の値が "world" です。

3. 共用プロセスを作成し、print **-p** を使用して共用プロセスに書き込み、read **-p** を使用して共用プロセスから入力を読み取るには、次のように入力します。

```
(read; print "hello $REPLY")
print -p "world"
read -p line
```



*line* 変数の値は "hello world" です。

4. 入力行のコピーをコマンドとしてヒストリー・ファイルに保存するには、次のように入力します。

```
read -s line < input_file
```

*input\_file* に "echo hello world" が入ると、"echo hello world" がコマンドとしてヒストリー・ファイルに保管されます。

## 関連情報

**printf** コマンド。

**ksh** コマンド。

---

## readlvcopy コマンド

### 目的

論理ボリュームの特定のミラー・コピーを読み取ります。

### 構文

```
readlvcopy -d device [ -c copy | -C copy | -b ] [ -n number_of_blocks ] [ -o outfile ] [ -s skip ] [ -S seek ]
```

### 説明

#### フラグ

|                                   |                                                                                                                                    |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| <b>-d</b> <i>device</i>           | 読み取り元の論理ボリューム・デバイス・ファイル。                                                                                                           |
| <b>-c</b> <i>copy</i>             | 読み取り元の要求されたミラー・コピー。データの 1 番目のコピー、2 番目のコピー、3 番目のコピーに有効な値は、1、2、または 3。データは、論理区画に期間切れというマークが付けられている場合でも読み取られます。デフォルトは、データの 1 番目のコピーです。 |
| <b>-C</b> <i>copy</i>             | 読み取り元の要求されたミラー・コピー。データの 1 番目のコピー、2 番目のコピー、3 番目のコピーに有効な値は、1、2、または 3。期間切れの論理区画は読み取られません。                                             |
| <b>-b</b>                         | オンライン・バックアップのマークが付けられたミラー・コピーを読み取りません。                                                                                             |
| <b>-n</b> <i>number_of_blocks</i> | 読み取る 128K ブロックの数。                                                                                                                  |
| <b>-o</b> <i>outfile</i>          | 宛先ファイル。デフォルトは <i>stdout</i> です。                                                                                                    |
| <b>-s</b> <i>skip</i>             | <i>device</i> にスキップする 128K ブロックの数。                                                                                                 |
| <b>-S</b> <i>seek</i>             | <i>outfile</i> にシークする 128K ブロックの数。                                                                                                 |

## 関連情報

**chlvcopy** コマンド。

---

## reboot または fastboot コマンド

### 目的

システムをリブートします。

### 構文

```
{ reboot | fastboot } [ -l ] [ -n ] [ -q ] [ -t mmddHHMM [ yy ] ]
```

### 説明

ほかのユーザーがシステムにログインしていなければ、**reboot** コマンドを使用してリブート操作を行うことができます。システムが動作中に複数のユーザーがシステムにログインしている場合には、**shutdown** コマンドを使用します。これらのコマンドはともに、故意にシステムを一時的に使用不可にしたことを示すエントリーをエラー・ログに書き込みます。

**reboot** コマンドを使用すると、**sync** コマンドでディスクを同期化し、システムがハードウェアの時刻機構を同期化しなおすなどの、ほかのシャットダウン・アクティビティーを行えるようになります。その後でリブート操作が始動されます。デフォルトでは、システムが始動するとディスクが自動的に検査されます。成功すると、システムはデフォルトの実行レベルになります。

リブートさせるには、機種に応じてリブート専用予約されたキーを押します。ほかのすべてが失敗したら、コンピューターをオフにし、ディスクの回転が停止するのを最低 60 秒待機して、再びオンにします。

**autorestart** 属性値が **true** であれば、電源が入ったとき、またはシステムのクラッシュ後、システムは自動的にリブートします。**autorestart** 属性の値を検出するには、**lsattr** コマンドを使用して **lsattr -D -l sys0** と入力します。デフォルト値は **true** です。**autorestart** 属性値を **false** にリセットするには、**chdev** コマンドを使用して次のように入力します。

```
chdev -l sys0 -a autorestart=false.
```

診断テストを行うと、システム ROM はブートストラップ・レコードを探します。システムはレコードが見つかるまでループします。一般に、ブートストラップ・レコードはディスクまたはハード・ディスクからブート・プログラムを読み取ります。デバッガーがカーネル内で構成されている場合は、最初にデバッガーが入力されますが、約 30 秒以内に何も入力されなければ、暗黙の **go** が実行されます。**go** は復帰でも実行できます。

一般に、**reboot** コマンドは、**syslog** コマンドを使用してリブート操作をログに記録し、**/var/adm/wtmp**、つまりログイン・アカウント・ファイルにシャットダウン・レコードを書き込みます。**-l**、**-n**、または **-q** フラグが指定されていれば、これらのアクションは禁止されます。

**fastboot** コマンドは、**reboot** コマンドを呼び出すことによってシステムをリブートさせます。**fsck** コマンドがシステムの始動時に実行され、ファイルシステムを検査します。このコマンドには BSD 互換性があります。

### フラグ

- l** リブートをログに記録すること、またはアカウント・ファイルにシャットダウン・レコードを書き込むこともしませんが、**-l** フラグは、アカウント・ファイルの更新を抑制しません。**-n** フラグおよび **-q** フラグは **-l** を暗黙指定します。
- n** **sync** コマンドを実行しません。このフラグを使うと、ファイルシステムが損傷することがあります。

- q** 最初に実行中のプロセスをシャットダウンせずにリブートします。  
注: **-q** フラグを使用すると、ファイルシステムの同期化は起こりません。ファイルシステムを同期化させたい場合には、**sync** コマンドを手動で実行するか、または **shutdown -r** コマンドを実行します。
- t** システムを直ちにシャットダウンし、その後、指定された日付にシステムを再始動します。有効な日付のフォーマットは以下のとおりです。

*mmddHHMM* [ *yy* ]

それぞれの意味は次のとおりです。

- mm* 月を指定します。
- dd* 日を指定します。
- HH* 時間を指定します。
- MM* 分を指定します。
- yy* 年を指定します (オプション)。

## 例

リブートをログに記録せずにシステムをシャットダウンするには、以下のように入力します。

```
reboot -l
```

## ファイル

- /etc/rc** システム始動スクリプトを指定します。
- /var/adm/wtmp** ログイン・アカウント・ファイルを指定します。

## 関連情報

**chdev** コマンド、**fasthalt** コマンド、**fsck** コマンド、**halt** コマンド、**lsattr** コマンド、**rc** コマンド、**shutdown** コマンド、**sync** コマンド、**syslogd** コマンド。

**utmp**、**wtmp**、**failedlogin** ファイル・フォーマット。

---

## recreatevg コマンド

### 目的

指定されたディスク・セットに既に存在しているボリューム・グループを再作成します。このボリューム・グループをインポートし、オンに構成変更します。

### 構文

```
recreatevg [ -y VGname ] [ -p ] [ -f ] [ -Y Lv_Prefix | -l LvNameFile ] [ -L Label_Prefix ] [ -n ]  
PVname...
```

### 説明

**recreatevg** コマンドは、特定のボリューム・グループに属している別のディスク・セットから複製されたディスク・セットのボリューム・グループを再作成します。このコマンドにより、ディスク複製プロセスによって発生する論理ボリューム管理 (LVM) のデータ構造および ID が重複するという問題が解決されま

す。このコマンドは、メンバー・ディスクに新しい物理ボリューム ID (PVID) を割り当てます。PVID もディスク複製機能によって複製されるからです。同じように、複製された論理ボリューム・メンバーには、ユーザー指定の接頭部を含む新しい名前が付けられます。

**注:**

1. **recreatevg** コマンドは、コマンド行に指定されていない物理ボリュームの一部または全部が存在するすべての論理ボリュームを除去します。ミラーリングされた論理ボリュームは例外とすることができます (**-f** フラグ参照)。
2. コマンド行に指定されたディスクにファイルシステムのログ論理ボリュームが存在していないと、**recreatevg** コマンドは警告を出します。
3. 入力リストと、ボリューム・グループ・ディスクリプター域 (VGDA) からコンパイルされたリストが一致しないと、**recreatevg** コマンドは失敗します。
4. リスト中のディスク・セットの VGDA データは、整合性のあるものでなければなりません。VGDA の問題は、**recreatevg** コマンドを使用しても修正されません。
5. 同時可能なボリューム・グループを再作成する場合、**recreatevg** コマンドが完了したとき、そのボリューム・グループはオンに変更されません。新しいボリューム・グループは手動でオンに変更する必要があります。

## フラグ

- f** ディスクのサブセットからボリューム・グループ (VG) を再作成します。ディスクのこのサブセットに完全に存在するディスクおよび論理ボリューム (LV) のみが、再作成される VG に存在します。オリジナルの VG からの他の全ディスクおよび LV は、再作成対象の VG の中では削除されることとなります。
- ミラーリングされた LV の場合、削除対象のディスク上に割り当てられた物理区画に伴う LV ミラー・コピーのみ削除されます。このため、もっと少ないミラー・コピーを使って、ミラーリングされた LV を再作成できますが、再作成できるのは、それらのコピーの最低 1 つがディスクのサブセット上に完全に存在する場合に限定されます。
- l *LvNameFile*** 論理ボリューム名を *LvNameFile* で指定された名前に変更します。エントリーは LV:NEWLV1 フォーマットになっていなければなりません。 *LvNameFile* に含まれていないすべての論理ボリュームは、デフォルトのシステム生成名で再作成されます。 NEWLV1 名を *LvNameFile* スタンザ内の LV 名と同じにし (LV:NEWLV1)、論理ボリュームを同じ名前のままにすることができます。
- L *Label\_Prefix*** 再作成される VG 上の論理ボリュームのラベルをこの接頭部に変更します。マウント・ポイントの簡単な修正だけではスタンザを一意的に定義できない場合は、*/etc/filesystems* スタンザを手で修正する必要があります。 / (スラッシュ) を *Label\_Prefix* に指定すると、論理ボリューム内のラベルは変更されません。
- n** **recreatevg** の後でボリューム・グループをインポートするが、オフに構成変更することを指定します。デフォルトがインポートされ、オンに構成変更されます。
- p** 新規 PVID の自動生成を使用不可にします。 **-p** フラグを使用する場合は、重複した PVID がシステムに含まれていないことを確認する必要があります。ハードウェア・ミラーリングされたすべてのディスクの PVID を固有な値に変更する必要があります。

|                            |                                                                                                                                                                                                                                                                                 |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-y</b> <i>VGname</i>    | ボリューム・グループ名が自動生成されないように指定できるようにします。ボリューム・グループ名はシステム内で固有でなければなりません。ボリューム・グループ名は 1 から 15 文字の範囲にすることができます。ボリューム・グループ名は、他のデバイスのデバイス構成データベースの PdDv クラスに既に定義されている接頭部で始めることはできません。新規のボリューム・グループ名は標準出力に送信されます。                                                                          |
| <b>-Y</b> <i>Lv_Prefix</i> | 再作成されるボリューム・グループ上の論理ボリュームをこの接頭部で名前変更できるようにします。接頭部と論理ボリューム名を合わせた長さは、15 文字以下でなければなりません。この長さが 15 文字を超えると、論理ボリュームの名前はデフォルトの名前に変更されます。この名前は、デバイス構成データベースの PdDv クラスに既に定義されている接頭部で始めることもできず、また、既に別のデバイスで使用されている名前にすることもできません。 <i>Lv_Prefix</i> に NA を指定すると、すべての論理ボリューム名は変更されないままで残ります。 |

## セキュリティ

アクセス制御: このコマンドを実行するには、root 権限を持っていないければなりません。

## 例

- 3 つの物理ボリュームが含まれたボリューム・グループを再作成するには、次のように入力します。

```
recreatevg hdisk1 hdisk2 hdisk3
```

hdisk1、hdisk2、および hdisk3 上のボリューム・グループは、自動生成された名前 (表示されます) で再作成されます。

- hdisk1 上のボリューム・グループを新規名 **testvg** で再作成するには、次のように入力します。

```
recreatevg -y testvg hdisk1
```

- hdisk14 上のボリューム・グループを再作成し、そのボリューム・グループ内のすべての論理ボリュームを再作成し、それらの論理ボリュームを接頭部 **newlv** で名前変更するには、次のように入力します。

```
recreatevg -Y newlv hdisk14
```

## ファイル

**/usr/sbin**            **recreatevg** コマンドが含まれているディレクトリー。

## 関連情報

**chvg** コマンド、 **chlv** コマンド。

---

## recsh コマンド

### 目的

リカバリー・シェルを起動します。

### 構文

```
recsh
```

## 説明

**libc.a** ライブラリーが移動されたかまたは名前変更された場合、ユーティリティをロードして実行するためにシステムが使用可能な **libc.a** ライブラリーがないため、シェルからエラー・メッセージ **Killed** が表示されることになります。 **recsh** コマンドは、意図せずに **libc.a** ライブラリーが移動された場合に、これを名前変更する機能を提供するリカバリー・シェルを起動します。リカバリー・シェルは、システムとともに出荷される代替の **libc.a** ライブラリーを使用します。

注: これはリカバリー・シェルであり、ユーザーは **recsh** をデフォルトのシェルとして使用しないでください。

## 例

1. 誤って **libc.a** が名前変更されると、システムは不安定状態になり、いかなるユーティリティの実行も不可能になります。この時点でリカバリーするには、次のように入力します。

```
recsh; cp -p libc.a.new /usr/lib/libc.a; exit
```

## 位置

**/usr/bin/recsh**

## ファイル

**/usr/bin/recsh**                   リカバリー・シェルへのパス名を指定します。

## 関連情報

AIX 5L バージョン 5.3 コマンド・リファレンス 第 1 巻の **bsh** コマンド。

AIX 5L バージョン 5.3 コマンド・リファレンス 第 3 巻の **ksh** コマンド。

AIX 5L バージョン 5.3 コマンド・リファレンス 第 5 巻の **sh** コマンド。

---

## redefinevg コマンド

### 目的

デバイス構成データベース内の指定されたボリューム・グループの物理ボリュームの集合を再定義します。

### 構文

```
redefinevg { -d Device | -i Vgid } VolumeGroup
```

### 説明

通常操作時には、デバイス構成データベースは物理ボリューム上にあるリザーブされた領域内の論理ボリューム・マネージャー (LVM) 情報と一致しています。デバイス構成データベースと LVM の間に不整合が生じると、**redefinevg** コマンドは、どの物理ボリュームが指定のボリューム・グループに属しているかを判別し、この情報をデバイス構成データベースに再入力します。**redefinevg** コマンドは、システムに接続されているすべての構成済み物理ボリュームのリザーブされた領域を読み取ることによって、不整合がないか検査します。

注: このコマンドを使用するには、root ユーザー権限を持っているか、または **system** グループのメンバーでなければなりません。

## フラグ

- d** *Device*           ボリューム・グループ ID、*Vgid* が、指定された物理ボリューム・デバイスから読み取られます。再定義対象のボリューム・グループに属する物理ボリュームであれば、どのボリュームの *Vgid* でも指定できます。
- i** *Vgid*             再定義されるボリューム・グループのボリューム・グループ識別番号。

## 例

デバイス構成データベースの rootvg 物理ボリュームを再定義するには、以下のようなコマンドを入力します。

```
redefinevg -d hdisk0 rootvg
```

## ファイル

`/usr/sbin/synclvodm`                   **synclvodm** コマンドが入っています。

## 関連情報

**varyonvg** コマンド、 **varyoffvg** コマンド、 **lsvg** コマンド。

---

## reducevg コマンド

### 目的

ボリューム・グループから物理ボリュームを除去します。物理ボリュームがボリューム・グループからすべて除去されると、そのボリューム・グループは削除されます。

### 構文

```
reducevg [ -d ] [ -f ] VolumeGroup PhysicalVolume ...
```

### 説明

**注意:** ボリューム・グループが並行モードにある間は、**reducevg** コマンドを使用できます。ただし、ボリューム・グループが並行モードにあり、終了結果がボリューム・グループの削除であるときにこのコマンドを実行すると、**reducevg** コマンドは失敗します。

**reducevg** コマンドは、*PhysicalVolume* パラメーターで指定された単一または複数の物理ボリュームを *VolumeGroup* から除去します。ボリューム・グループ内のすべての物理ボリュームを削除すると、ボリューム・グループも除去されます。ボリューム・グループを縮小する前に、オンに変更しておかなければなりません。

**reducevg** コマンドを始動する前に、**rmlv** コマンドまたは **-d** フラグを使用して、*PhysicalVolume* パラメーターで指定した、物理ボリューム上に存在するすべての論理ボリュームを除去しなければなりません。



## 注:

1. このコマンドを使用するには、root ユーザー権を持っているか、または **system** グループのメンバーでなければなりません。
2. 最初に **reducevg** *VolumeGroup PhysicalVolume* を実行しないと、ディスクがシステムから除去されることがあります。ディスクは除去されてもまだ VGDA のメモリーに入っていますが、*PhysicalVolume* 名は、もはや存在していないか、または割り当てし直されています。この存在しないディスクに対する参照を除去するために、まだ **reducevg** を使用できますが、ディスク名: **reducevg** *VolumeGroup PVID* ではなく、物理ボリューム ID (PVID) を使用します。
3. スナップショット・ボリューム・グループには **reducevg** コマンドは使用できません。

AIX 5.3 上で作成されて **varyonvg -M** フラグを指定せずにオンに変更されたボリューム・グループの場合、**reducevg** は、残った物理ボリュームの共通最大転送サイズと一致することが必要であれば、ボリューム・グループの論理トラック・グループ・サイズを動的に増やします。

Web-based System Manager (wsm) の ボリューム アプリケーションを使用してボリューム特性を変更できません。

このコマンドは、System Management Interface Tool (SMIT) の **smit reducevg** 高速パスを使用しても実行できます。

## フラグ

- d 既存の論理ボリューム区画の割り当てを解除して、結果的に空になった論理ボリュームを、指定された物理ボリュームから削除します。-f フラグを追加しない場合は、ユーザーの確認が必要です。  
**注意:** **reducevg** コマンドに -d フラグを付けると、ボリューム・グループから物理ボリュームが除去される前に、物理ボリューム上のすべての論理ボリューム・データが自動的に削除されます。論理ボリュームが複数の物理ボリュームにまたがっている場合は、それらの物理ボリュームのいずれかを除去すると、論理ボリューム全体の整合性が損なわれることがあります。
- f -d フラグを使用したときの、ユーザーへの確認要求を排除します。

## 例

1. ボリューム・グループ **vg01** から物理ボリューム **hdisk1** を除去するには、次のように入力します。  

```
reducevg vg01 hdisk1
```
2. ユーザーへの確認を行わずに、ボリューム・グループ **vg01** から物理ボリューム **hdisk1** および存在するすべての論理ボリュームを除去するには、次のように入力します。 **注意:** **reducevg** コマンドに -d フラグを付けると、すべての論理ボリュームを自動的に削除してから、物理ボリュームが除去されます。

```
reducevg -d -f vg01 hdisk1
```

物理ボリューム **hdisk1** および常駐するすべての論理ボリュームが除去されます。

## ファイル

**/usr/sbin/reducevg**  
**/tmp**

**reducevg** コマンドが常駐するディレクトリー。  
コマンドの実行中に一時ファイルが保存されるディレクトリー。

## 関連情報

**extendvg** コマンド、 **rmlv** コマンド。

オペレーティング・システムおよびデバイスのマネージの論理ボリューム・ストレージのセクション。

Web-based System Manager のインストールについては、 *AIX 5L Version 5.3 Web-based System Manager 管理ガイド* の第 2 章 インストールおよびシステム要件 のセクションを参照してください。

オペレーティング・システムおよびデバイスのマネージの System Management Interface Tool のセクション。

---

## refer コマンド

### 目的

文書内で参考資料を見つけ、その資料を文書に挿入します。

### 構文

```
refer [ -b ] [ -e ] [ -P ] [ -S ] [ -a [ Number ] ] [ -B Label.Macro ] [ -c Keys ] [ -f Number | -k Label | -l Letter, Digit ] [ -n ] [ -p Reference ] [ -s Keys ] [ File ... ]
```

### 説明

**refer** コマンドは、**nroff** または **troff** コマンドのプリプロセッサです。**refer** コマンドは、脚注または巻末の注に関する参照を検索してフォーマットします。また、このコマンドは、適切な書式で入力されたスタンドアロン参考文献一覧などのデータに索引を付け、検索し、ソートして出力するように設計された一連のプログラムの基礎となります。

引用が不完全であってもキーワードが十分に正確であれば、**refer** コマンドは、参考文献データベースを検索し、タイトル、著者、定期刊行物などのどこかにこれらのキーワードを持つ参照がないか調べます。入力ファイル (そうでなければ標準入力) は、`.[` (ピリオド、左大括弧) と `.]` (ピリオド、右大括弧) 区切り文字で囲まれた行を除き、標準出力にコピーされます。区切り文字で囲まれた行は、キーワードを含むものと見なされ、参考文献データベースからの情報で置換されます。ユーザーは、別のデータベースを検索したり、特定のフィールドを上書きしたり、新規フィールドを追加することもできます。ソースがどこであれ、参照データは、**troff** コマンド・ストリングのセットに割り当てられます。**ms** マクロ・パッケージなどのマクロ・パッケージは、これらの文字列から生成された完全な参照テキストを印刷します。デフォルトでは、参照には脚注番号のフラグが付きます。

独自の参照を使用する場合は、それらを例のセクションで説明するフォーマットにします。これらの参照にアクセスするには、**-p** フラグを使用するか、**REFER** 環境変数をそれらの参照ファイルに設定します。

**refer** コマンドを使用する前に、参照に対して **indxib** コマンドを実行すると、参照の検索速度を増すことができます。索引を付けなければ、線型検索が実行されます。**refer** コマンドをプリプロセッサ・コマンド (**eqn**、**neqn**、または **tbl** コマンド) のいずれかと一緒に使用する場合は、パイプ接続を介して渡されるデータ量を最小に抑えるために、最初に **refer** コマンドを発行しなければなりません。

**注:** 参照ファイルを編集する場合、そのファイルに対して **indxib** コマンドを再発行しなければなりません。**indxib** コマンドを使用しない場合は、その参照ファイルに関連する **.ia**、**.ib**、**.ic**、および **.ig** ファイルをすべて除去してください。これらのファイルを除去しないと、**refer** コマンドから「too many hits」というエラー・メッセージが表示されます。

**refer** コマンドおよびそれに関連するプログラムは、ブランク行で区切られたレコードからなる参照ファイルから入力されることを予期します。レコードとは、それぞれに 1 種類の情報が入っているフィールド (行) の集合体です。フィールドは、先頭の % (パーセント記号) の後にキー文字、スペース文字、およびフィールドの内容が続く行で始まり、最後にフィールドの内容を持つ行で開始し、% (パーセント記号) で始まる次の行まで続きます。フィールドの出力順序とフォーマットは、**nroff** および **troff** コマンド用に指定されたマクロ (脚注および巻末の注用)、または **roffbib** コマンド用に指定されたマクロ (スタンドアロン参考文献一覧用) で制御されます。最も一般的なキー文字およびそれらに対応するフィールドのリストについては、**addbib** コマンドのセクションを参照してください。

## フラグ

- b** ベア・モード: テキスト内にフラグを入れません (数字もラベルも)。
- e** 参照を見つけた場所に置いておく代わりに、次のフォーマットのシーケンスを見つけるまで累積します。  
.[  
\$LIST\$  
.]
- P** 次に、それまで収集されたすべての参照を書き出します。参照シグナルの前ではなく、その後に句読点を入れます。句読点は場所により異なり、**refer** メッセージ・カタログ内で定義されます。
- S** 参照を自然科学または社会科学フォーマットで生成します。
- a Number** 最初から数えて指定数の著者名を反転させます (J. A. Jones ではなく Jones, J. A.)。 *Number* 変数を省略すると、すべての作成者名が反転します。
- B Label.Macro** 参考文献一覧モードを指定します。ブランク行で区切られたレコードからなるファイルを選んで、そのファイルを **troff** コマンド入力にします。指定ラベルは、指定マクロになり、デフォルトでは *Label* 変数値は **%X**、*Macro* 変数値は **.AP** (注釈パラグラフ) となります。
- c Keys** キー文字が *Keys* 変数内にあるフィールドを SMALL CAPS を使用して大文字に変換します。例えば、Jack は JACK になります。
- f Number** 脚注番号をデフォルトの 1 のではなく指定された番号に設定します。数値ではなくラベルを使用する場合、このフラグは効果がありません。 **-k** フラグおよび **-l** フラグのセクションを参照してください。
- k Label** 番号の参照を使う代わりに、**%Label** で始まる参照データ行内で指定されたラベルを使用します。デフォルトでは、*Label* 変数値は **L** です。
- l Letter, Digit** 番号の参照を使う代わりに、主著者の姓と出版年度からなるラベルを使用します。姓のうち前の方の指定文字と、日付のうち後の方の指定数字だけを使用します。 *Letter* 変数または *Digit* 変数を省略すると、それぞれ、名前全体または日付全体が使用されます。
- n** デフォルトの **/usr/share/dict/papers/ln** ファイルを検索しません。 **REFER** 環境変数を設定すると、デフォルト・ファイルの代わりに指定したファイルが検索されます。この場合には、 **-n** フラグは影響を与えません。
- p Reference** *Reference* 変数を検索対象の参照ファイルとして扱います。デフォルト・ファイルは最後に検索されます。

**-s** *Keys*

キー文字が *Keys* 変数文字列で指定されたフィールド別に参照をソートします。それに  
応じてテキスト内の参照番号を付け直します。このフラグは **-e** フラグを暗黙指定しま  
す。*Keys* 変数で指定されたキー文字の後に数値を指定して、このようなフィールドをい  
くつ使用するかを示すことができます。この場合、**q +** (プラス記号) はきわめて大きな  
数値を示します。デフォルト値は **AD** で、最初に主著者で、次に日付でソートします。  
例えば、すべての著者、次にタイトルでソートするときは、**-sA+T** を入力します。

参考文献一覧フィールドの行末にブランク・スペースを付けると、レコードが正しくソ  
ートおよび反転されないので注意してください。多量の参照をソートすると、メモリ  
ー・ダンプが起こることがあります。

## 例

以下は、**refer** コマンドのエントリー例です。

%AM.E. Lesk

%T Some Applications of Inverted Indexes on the UNIXSystem

%B UNIXProgrammer's Manual

%V 2b

%I Bell Laboratories

%C Murray Hill, NJ

%D 1978

## ファイル

**/usr/share/dict/papers/Ind**  
**/usr/sbin/refer**

デフォルトの参照ファイルが入っています。  
関連プログラムが入っています。

## 関連情報

**eqn** コマンド、**addbib** コマンド、**indxbib** コマンド、**lookbib** コマンド、**neqn** コマンド、**nroff**  
コマンド、**roffbib** コマンド、**sortbib** コマンド、**tbl** コマンド、**troff** コマンド。

---

## refile コマンド

### 目的

フォルダー間でファイルを移動します。

### 構文

**refile** [ **-src** *+Folder* ] [ **-draft** ] [ **-file** *File* ] [ *Messages* ] [ **-nolink** | **-link** ] [ **-nopreserve** | **-preserve**  
] *+Folder* ...

## 説明

**refile** コマンドは、フォルダー間でメッセージを移動します。ソース・フォルダーを指定しなければ、

**refile** コマンドは現行フォルダーをソースとして使用します。存在しない宛先フォルダーを指定すると、システムはそのフォルダーを作成するかどうかを照会します。

また、**refile** コマンドは、あるフォルダーから別のフォルダーにメッセージをコピーします。デフォルトでは、メッセージを移動するとき、システムはそのメッセージのコピーをオリジナル・フォルダーに保持しません。コピーを残すには、**-preserve** フラグを使用します。

## フラグ

**-draft**

メール・ディレクトリーから現行ドラフト・メッセージをコピーします。

**-file** *File*

指定されたファイルをコピーします。ファイルは有効なメッセージ形式でなければなりません。新規メッセージを正しくフォーマットしファイルするには、**inc** コマンドを使用します。指定されたフォルダーにメッセージをコピーします。任意の数のフォルダーを指定できます。コマンド構文、使用可能なスイッチ (トグル)、およびバージョン情報をリストします。

**+Folder**

**-help**

**注:** MH の場合は、このフラグ名を省略することはできません。

**-link**

*Messages*

メッセージをコピーした後、ソース・フォルダー、またはファイルに残します。

コピー対象となるメッセージを指定します。複数のメッセージ、メッセージの範囲、または単一メッセージを指定できます。以下のリファレンスを使用してメッセージを指定します。

*Number* メッセージの数。

*Sequence*

ユーザーが指定するメッセージのグループ。指定できる値は、以下のとおりです。

**all** フォルダー内のすべてのメッセージ。

**cur** または **.** (ピリオド)

現行メッセージ。これはデフォルトです。

**first** フォルダー内の最初のメッセージ。

**last** フォルダー内の最後のメッセージ。

**next** 現行メッセージの次のメッセージ。

**prev** 現行メッセージの前のメッセージ。

**/DT>** **-link** フラグと **all** フラグを同時に指定すると、現行フォルダーの現行メッセージは変化しません。そうでない場合は、メッセージを指定すると、リファイルされたメッセージが現行メッセージになります。

**-nolink**

メッセージをコピーした後でソース・フォルダーまたはファイルから除去します。このフラグがデフォルトです。

**-nopreserve**

コピー対象となるメッセージの番号を付け直します。番号の付け直しは、宛先フォルダーの最後のメッセージの番号より 1 つ大きい番号で始まります。このフラグがデフォルトです。

**-preserve**

コピーしたメッセージ番号を保存します。その番号の付いたメッセージが既に存在すると、

**-src** **+Folder**

**refile** コマンドはエラー・メッセージを出し、フォルダーの内容は変化しません。ソース・フォルダーを識別します。デフォルトでは、システムは現行フォルダーを使用します。

## プロフィール・エントリー

以下のエントリーは、*UserMhDirectory/.mh\_profile* ファイルの一部です。

Current-Folder: デフォルトの現行フォルダーを設定します。  
Folder-Protect: 新規フォルダー・ディレクトリーの保護レベルを設定します。  
Path: *UserMhDirectory* を指定します。  
rmmproc: メッセージをフォルダーから除去するときに使用するプログラムを指定します。

## 例

1. 現行フォルダーから現行メッセージをリファイルして、*meetings* という名前の新規フォルダーに入れるには、以下のように入力します。

```
refile +meetings
```

システムは次のようなメッセージで応答します。

```
Create folder "/home/jeanne/Mail/meetings"?
```

フォルダーを作成するには *y* と入力します。オリジナル・メッセージのコピーは現行フォルダーに保存されません。

2. 現行フォルダーから現行メッセージを *meetings* フォルダーにコピーするには、以下のように入力します。

```
refile -link +meetings
```

オリジナル・メッセージはそのまま現行フォルダー内に残ります。

3. 現行メッセージ・ドラフトを *test* フォルダーにリファイルするには、以下のように入力します。

```
refile -draft +test
```

メッセージ・ドラフトのコピーは、現行フォルダー内に保存されません。

4. 現行フォルダーから現行メッセージを複数のフォルダーにリファイルするには、以下のように入力します。

```
refile +tom +pat +jay
```

メッセージのコピーは現行フォルダーには保存されません。

## ファイル

*\$HOME/.mh\_profile* MH ユーザー・プロフィールを設定します。  
*/usr/bin/refile* **refile** コマンドが入っています。

## 関連情報

**folder** コマンド、**folders** コマンド。

**.mh\_alias** ファイル・フォーマット、**.mh\_profile** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージのメール・アプリケーションのセクション。



---

## refresh コマンド

### 目的

単一またはグループのサブシステムのリフレッシュを要求します。

### 構文

```
refresh [ -h Host ] { -g Group | -p SubsystemPID | -s Subsystem }
```

### 説明

**refresh** コマンドは、サブシステムに送られるサブシステム・リフレッシュ要求をシステム・リソース・コントローラーに送信します。リフレッシュ・アクションは、サブシステムによって異なります。

注: サブシステムの通信手法がシグナルの場合は、**refresh** コマンドは成功しません。

### フラグ

- |                               |                                                                                                                                                                                                                                                                                                          |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-g</b> <i>Group</i>        | リフレッシュの対象となるサブシステム・グループを指定します。 <i>Group</i> 名がサブシステムのオブジェクト・クラス内にいていなければ、 <b>refresh</b> コマンドは失敗に終わります。                                                                                                                                                                                                  |
| <b>-h</b> <i>Host</i>         | このリフレッシュ・アクションの要求先となる外部の <i>Host</i> コンピューターを指定します。ローカル・ユーザーは、「root」として実行している必要があります。リモートのシステム・リソース・コントローラー要求を受け入れるようリモート・システムを構成しておく必要があります。つまり、 <b>srcmstr</b> デーモン ( <i>/etc/inittab</i> を参照) を <b>-r</b> フラグで開始し、かつ、リモート要求を許可するよう <i>/etc/hosts.equiv</i> ファイルまたは <b>.rhosts</b> ファイルを構成しておく必要があります。 |
| <b>-p</b> <i>SubsystemPID</i> | リフレッシュの対象となるサブシステムの特定のインスタンスを指定します。                                                                                                                                                                                                                                                                      |
| <b>-s</b> <i>Subsystem</i>    | リフレッシュの対象となるサブシステムを指定します。 <i>Subsystem</i> 名には、実際のサブシステム名、またはサブシステムの同義語名が使えます。 <i>Subsystem</i> 名がサブシステムのオブジェクト・クラスに入っていないと、 <b>refresh</b> コマンドは失敗に終わります。                                                                                                                                               |

### 例

1. tcpip などのグループをリフレッシュするには、以下のように入力します。

```
refresh -g tcpip
```

2. xntpd などのサブシステムをリフレッシュするには、以下のように入力します。

```
refresh -s xntpd
```

### ファイル

- |                                |                                   |
|--------------------------------|-----------------------------------|
| <i>/etc/objrepos/SRCsubsys</i> | SRC サブシステム構成オブジェクト・クラスを指定します。     |
| <i>/etc/services</i>           | インターネット・サービス用のソケットおよびプロトコルを定義します。 |
| <i>/dev/SRC</i>                | <b>AF_UNIX</b> ソケット・ファイルを指定します。   |
| <i>/dev.SRC-unix</i>           | 一時ソケット・ファイルの位置を指定します。             |

### 関連情報

**startsrc** コマンド、 **stopsrc** コマンド。



オペレーティング・システムおよびデバイスのマネージのシステム・リソース・コントローラーのセクション。

---

## refsrc コマンド

### 目的

指定したリソース・クラス内でリソースをリフレッシュします。

### 構文

```
refsrc [-h] [-TV] resource_class
```

### 説明

**refsrc** コマンドは、指定したリソース・クラス内でリソースをリフレッシュします。このコマンドを使用して、オペレーティング・システム・コマンド (例: **mkfs**) で変更できる構成がある場合に、RMC (リソース・モニターおよび制御) サブシステムにリソースの新しいインスタンスを検出させます。

このコマンドを使用して、RMC サブシステムにリソース・クラス内のリソースの構成を要求します。要求は、実際にはリンクされたリソース・マネージャーが実行します。

指定したリソース・クラス内でリソースをモニターするアプリケーションはどれでも、構成が更新されたとのイベントを受け取ります。

### フラグ

- h コマンドの使用ステートメント (使用法) を標準出力に書き込みます。
- T コマンドのトレース・メッセージを標準エラーに書き込みます。ソフトウェア・サービスの編成の使用の場合のみです。
- v コマンドの詳細メッセージを標準出力に書き込みます。

### パラメーター

*resource\_class*                      リソース・クラス名を指定します。

### セキュリティ

ユーザーは、**refsrc** を実行するために **refsrc** に指定した *Resource\_class* に対して読み取り許可が必要です。権限は、相手のシステムのアクセス制御リスト (ACL) ファイルに指定します。ACL ファイルとその変更方法については、「*RSCT: Administration Guide*」を参照してください。

### 終了状況

- 0 コマンドが正常に実行されました。
- 1 RMC でエラーが起きました。
- 2 コマンド行インターフェース (CLI) スクリプトでエラーが発生しました。
- 3 コマンド行に間違っただフラグが指定されました。
- 4 コマンド行に不正なパラメーターが指定されました。

- 5 不正なコマンド行入力に基づく RMC により、エラーが発生しました。

## 環境変数

### CT\_CONTACT

CT\_CONTACT 環境変数をホスト名または IP アドレスに設定すると、コマンドは指定されたホストの RMC (リソース・モニターおよび制御) デーモンと連絡します。この環境変数を指定しないと、コマンドは、そのコマンドが実行されているローカル・システムの RMC デーモンと連絡します。コマンドにより表示または変更されるリソースまたはリソース・クラスは、接続が確立されたシステム上で探されます。

### CT\_IP\_AUTHENT

CT\_IP\_AUTHENT 環境変数が存在する場合、RMC デーモンは、IP ベース・ネットワーク認証を使用して、CT\_CONTACT 環境変数が設定されている IP アドレスで指定されたシステム上の RMC デーモンに連絡します。CT\_IP\_AUTHENT は、CT\_CONTACT が IP アドレスに設定されている場合にのみ意味を持ちます。ドメイン・ネーム・システム (DNS) サービスには依存しません。

### CT\_MANAGEMENT\_SCOPE

RMC デーモンがリソースおよびリソース・クラスをモニターし、制御するためのセッションに使用する管理有効範囲を指定します。管理有効範囲は、リソースおよびリソース・クラスをモニターし、制御する対象となる、可能なターゲット・ノードのセットを決定します。有効な値は以下のとおりです。

- 0 *local* 有効範囲を指定します。
- 1 *local* 有効範囲を指定します。
- 2 *peer domain* (ピア・ドメイン) 有効範囲を指定します。
- 3 *management domain* (管理ドメイン) 有効範囲を指定します。

この環境変数を設定しないと、*local* 有効範囲が使用されます。

## 実装上の固有な条件

このコマンドは、AIX 用の Reliable Scalable Cluster Technology (RSCT) ファイルセットの一部です。

## 標準出力

**-h** フラグが指定されている場合は、このコマンドの使用状況の説明文が標準出力に書き込まれます。

コマンド出力とすべての詳細メッセージは、標準出力に書き込まれます。

## 標準エラー

トレース・メッセージはすべて、標準エラーに書き込まれます。

## 例

- クラス IBM.FileSystem 内のリソースの構成をリフレッシュするには、次のように入力します。

```
refsrc IBM.FileSystem
```

## 位置

```
/usr/sbin/rsct/bin/refsrc
```

## 関連情報

資料: 「*RSCT: Administration Guide*」 (RMC 操作に関する情報)

コマンド: **lsrsrc**、**lsrsrcdef**

情報ファイル: **rmccli** (RMC 関連コマンドに関する一般情報)

---

## refsensor コマンド

### 目的

Resource Monitoring and Control (RMC) サブシステムに定義されているセンサーをリフレッシュします。

### 構文

```
refsensor [-a | -n host1[,host2...] | -N { node_file | "-" } ] [-h ] [-v | -V] sensor_name [ attr1=value1 [ attr2=value2 ] ... ]
```

### 説明

**refsensor** コマンドは、リソース・モニターおよび制御 (RMC) サブシステムに対して定義されているセンサー・リソースをリフレッシュします。センサーは、モニターが可能な属性を含む RMC リソースです。

**refsensor** を使用してセンサーをリフレッシュするには、2 種類の方法があります。1 つはセンサー・リソースに対して定義されているセンサー・コマンドを実行する方法、もう 1 つは特定のセンサー属性に対して値を指定する方法です。**refsensor** が正常に実行されるためには、センサーがモニター中でなければなりません。

**refsensor** によって特定のセンサー属性を更新するには、1 つ以上の *attr=value* パラメーターを指定します。更新されるのは指定された属性だけです。それ以外のセンサー属性は更新されません。パラメーターとして指定できるセンサー属性は、次のとおりです。

|                |                            |
|----------------|----------------------------|
| <b>Float32</b> | このセンサー・リソースの float32 型の属性。 |
| <b>Float64</b> | このセンサー・リソースの float64 型の属性。 |
| <b>Int32</b>   | このセンサー・リソースの int32 型の属性。   |
| <b>Int64</b>   | このセンサー・リソースの int64 型の属性。   |
| <b>Quantum</b> | このセンサー・リソースの quantum 型の属性。 |
| <b>String</b>  | このセンサー・リソースの string 型の属性。  |
| <b>Uint32</b>  | このセンサー・リソースの uint32 型の属性。  |
| <b>Uint64</b>  | このセンサー・リソースの uint64 型の属性。  |

例えば、**Sensor1** というセンサーについて、センサー属性 **Int32** と **Float32** だけを更新するには、次のように入力します。

```
refsensor Sensor1 Int32=45 Float32=7.8
```

それ以外のセンサー属性は更新されません。

センサー・コマンドの実行用に間隔が定義されている場合、**refsensor** コマンドを実行しても、それには影響しません。つまり、モニター対象のセンサーが 60 秒ごとに更新されている場合に **refsensor** を実行しても、インターバル・タイマーが 60 秒にリセットされるわけではありません。

**refsensor** コマンドは、どのノードでも実行されます。ドメイン内のすべてのノードで **refsensor** を実行したい場合は、**-a** フラグを使用します。ドメイン内の一部のノードで **refsensor** コマンドを実行したい場合には、**-n** フラグを使用します。

## フラグ

- a**      ドメイン内のすべてのノード上の、指定された名前と一致するセンサーをリフレッシュします。クラスタの有効範囲は、`CT_MANAGEMENT_SCOPE` 環境変数によって決まります。  
`CT_MANAGEMENT_SCOPE` が設定されていない場合、コマンドにとって有効な有効範囲になるまで、最初に管理ドメイン有効範囲 (存在する場合) を選択し、次にピア・ドメイン有効範囲 (存在する場合) を選択し、次にローカル有効範囲を選択します。コマンドは、最初に見つかった有効な有効範囲に対して、1 回実行されます。例えば、管理ドメインとピア・ドメインの両方が存在する場合に `CT_MANAGEMENT_SCOPE` を設定しない **refsensor -a** を実行すると、それは管理ドメインで実行されます。この場合、ピア・ドメインで実行するには `CT_MANAGEMENT_SCOPE` を 2 に設定します。
- h**      コマンドの使用ステートメント (使用法) を標準出力に書き込みます。
- n host**  
      センサーをリフレッシュするノードを指定します。デフォルトでは、ローカル・ノード上のセンサーがリフレッシュされます。このフラグは、管理ドメインまたはピア・ドメインでのみ使用します。
- N {node\_file | "-"}**  
      センサーの削除対象となるノードをリストしているファイルまたは標準入力を指定します。このフラグの使用が適切なのは、クラスタ・システム・マネージメント (CSM) またはピア・ドメイン・クラスタの中だけです。
- v | -V**  
      コマンドの詳細メッセージを標準出力に書き込みます。

## パラメーター

*sensor\_name*

リフレッシュするセンサーの名前を指定します。

*attr=value*

リフレッシュするセンサー属性と、その設定値を指定します。

## セキュリティ

**refsensor** を実行するためには、**IBM.Sensor** リソース・クラスに関する書き込みアクセス権が必要です。権限は、相手のシステムのアクセス制御リスト (ACL) ファイルに指定します。ACL ファイルおよびその変更方法について詳しくは、「*RSCT: Administration Guide*」を参照してください。

## 終了状況

- 0 コマンドが正常に実行されました。
- 1 フラグとパラメーターの間違った組み合わせが入力されました。
- 4 このセンサーはモニターされていないため、リフレッシュできません。
- 6 センサー・リソースが見つかりませんでした。
- n* RMC サブシステムから戻される可能性のあるその他のエラーに基づくものです。

## 環境変数

### CT\_CONTACT

**CT\_CONTACT** 環境変数をホスト名または IP アドレスに設定すると、コマンドは指定されたホストの RMC (リソース・モニターおよび制御) デーモンと連絡します。この環境変数を設定しないと、コマンドは、そのコマンドが実行されているローカル・システムの RMC デーモンと連絡します。コマンドにより表示または変更されるリソースまたはリソース・クラスは、接続が確立されたシステム上で探されます。

### CT\_IP\_AUTHENT

**CT\_IP\_AUTHENT** 環境変数が存在する場合、RMC デーモンは、IP ベース・ネットワーク認証を使用して、**CT\_CONTACT** 環境変数が設定されている IP アドレスで指定されたシステム上の RMC デーモンに連絡します。**CT\_IP\_AUTHENT** は、**CT\_CONTACT** が IP アドレスに設定されている場合にのみ意味を持ちます。ドメイン・ネーム・システム (DNS) サービスには依存しません。

### CT\_MANAGEMENT\_SCOPE

RMC デーモンがリソースおよびリソース・クラスをモニターし、制御するためのセッションに使用する管理有効範囲を指定します。管理有効範囲は、リソースおよびリソース・クラスをモニターし、制御する対象となる、可能なターゲット・ノードのセットを決定します。

有効な値は以下のとおりです。

- 0 *local* 有効範囲を指定します。
- 1 *local* 有効範囲を指定します。
- 2 *peer domain* (ピア・ドメイン) 有効範囲を指定します。
- 3 *management domain* (管理ドメイン) 有効範囲を指定します。

この環境変数を設定しないと、*local* 有効範囲が使用されます。

## 実装上の固有な条件

このコマンドは、AIX 用の Reliable Scalable Cluster Technology (RSCT) ファイルセットの一部です。

## 例

1. センサー **Sensor1** をリフレッシュして、それに対して定義されているセンサー・コマンドが実行されるようにするには、次のように入力します。

```
refsensor Sensor1
```

2. センサー **Sensor1** をリフレッシュして、**Int32** を **50** に設定し、**Float32** を **123.45** に、そして **String** を **"test input"** に設定するには、次のように入力します。

```
refsensor Sensor1 Int32=50 Float32=123.45 String="test input"
```

## 位置

`/usr/sbin/rsct/bin/refsensor`

## 関連情報

資料: 「*RSCT: Administration Guide*」 (ACL 許可ファイルに関する情報)

コマンド: **chsensor**、**lssensor**、**mksensor**、**rmsensor**

## regcmp コマンド

### 目的

パターンを C 言語の **char** 宣言にコンパイルします。

### 構文

```
regcmp [ - ] File [ File ... ]
```

### 説明

**regcmp** コマンドは、*File* 内のパターンをコンパイルして、出力の中のパターンをコンパイルして、出力を *File.i* ファイル内に配置し、また、**-** オプションを指定すると、出力は *File.c* ファイル内に配置されます。コンパイル結果のパターンは、**char** 宣言として初期化されます。*File* の各エントリは、後に 1 つまたは複数の空白と " " (二重引用符) で囲まれたパターンが続く C 変数名でなければなりません。

**regcmp** コマンドの出力は、C ソース・コードです。コンパイル結果の *File.i* ファイルは C プログラムに含めることができ、また、コンパイル結果の *File.c* ファイルは、**cc** コマンドへのファイル・パラメーターとして使用できます。

C 言語プログラムが **regcmp** コマンドの出力を使用する場合は、**regex** サブルーチンを使ってそれを文字列に適用しなければなりません。

たいていの場合、**regcmp** コマンドは C 言語プログラム内で **regcmp** サブルーチンを使う必要をなくすので、実行時間およびプログラム・サイズの節約になります。

### フラグ

- 出力を *File.c* ファイルに配置します。デフォルトでは、出力は *File.i* ファイルに配置されます。

### 例

1. *stdin1* 内のパターンおよび *stdin2* 内のパターンをコンパイルするには、以下のように入力します。

```
regcmp stdin1 stdin2
```

これにより、*stdin1.i* および *stdin2.i* ファイルが作成されます。

2. *stdin1.c* および *stdin2.c* ファイルを作成するには、以下のように入力します。

```
regcmp - stdin1 stdin2
```

**注:** どちらの例でも同じ *stdin1* および *stdin2* ファイルを使用すると想定すると、コンパイル結果の *stdin1.i* および *stdin1.c* ファイル、および、*stdin2.i* および *stdin2.c* ファイルは同じになります。

### ファイル

**/usr/ccs/bin/regcmp**

**regcmp** コマンドが入っています。

## 関連情報

**regcmp** または **regex** サブルーチン。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」のサブルーチンの概要のセクション。

---

## rembak コマンド

### 目的

印刷ジョブをリモート・サーバーのキューに送ります。

### 構文

```
rembak -S Server -P Queue [ -R ] [ -N Filter ] [ -L ] [ -p ] [ -q ] [ -x ] [ -# JobNumber ] [ -u  
UserName ] [ -X ] [ -o Option ] [ -T Timeout ] [ -C ] [ -D DebugOutputFile ] [ File ... ]
```

### 説明

**rembak** コマンドは、ジョブをリモート・サーバーに送ってキューに入れます。要求として印刷ジョブ、状況要求、ジョブ取り消し要求、または、リモート・キューイング・システムを抹消する要求が使えます。サーバーとキューのフラグが必要です。ほかのフラグはすべてオプションで、必要な実行内容によって指定します。

このコマンドを呼び出せるのは、**qdaemon** コマンドだけです。ユーザーがコマンド行から入力するコマンドではありません。印刷ジョブ要求の出し方、または System Manager Interface Tool (SMIT) を使用した印刷ジョブの要求方法については、**enq** コマンドのセクションを参照してください。

### フラグ

**-#JobNumber**

取り消したい *JobNumber* を指定します。

**-C**

最初に制御ファイルリストを送信します。**lpd** プロトコルにより、印刷ジョブを処理する 2 つのハンドシェーク・シーケンスが許可されます。デフォルトは、最初にデータ・ファイル、続いて制御ファイルの送信で構成されます。ほかのシーケンスでは、最初に制御ファイルを送信し、続いてデータ・ファイルを送信します。**-C** を指定すると、**rembak** は、最初に制御ファイルを送信し、続いてデータ・ファイルを送信します。

**-D DebugOutputfile**

**rembak** のデバッグ・オプションをオンにします。出力ファイルが指定されないか、または出力ファイルの作成またはそれへの書き込みに何らかの問題があると、デバッグ・オプションは無視されます。指定された出力ファイルが既に存在する場合は、その終わりに新規デバッグ出力が追加されます。

**-L**

リモート・キューからの長い (詳細) 状況要求を示します。



- N Filter** リモート・サーバーの機種を示します。フィルター名の指定は、`/etc/qconfig` ファイル内の `s_statfilter` 属性で行います。`filter` 変数には、以下のものがあります。
- /usr/lib/lpd/aixshort**  
サーバーが別の AIX マシンであることを示します。
- /usr/lib/lpd/aixv2short**  
サーバーが AIX バージョン 2 オペレーティング・システムの RT であることを示します。
- /usr/lib/lpd/bsdshort**  
サーバーが `bsd` コンピューターであることを示します。
- /usr/lib/lpd/attshort**  
サーバーが AT&T コンピューターであることを示します。
- o Option** リモート・サーバー上のバックエンドに送りたい `Option` を指定します。(これらの `Options` は、`rembak` コマンドから渡されます。)
- p** `rembak` が使用するポート範囲が 1023 より下のポートに制限されることを示します。
- P Queue** 印刷ジョブの送信先となるリモート・サーバーの `Queue` 名を指定します。
- q** リモート・キューからの短い (簡略な) 状況要求を示します。
- R** リモート・キューイング・システムを再始動します。  
注: オペレーティング・システムに要求を送信する場合は、`-R` フラグはサポートされません。`lpd` デモンはこのような要求をサポートしません。ほかのシステムとの互換性のためだけに、`-R` フラグはサポートされます。
- S Server** 印刷要求の送信先となるリモート印刷 `Server` 名を指定します。
- T Timeout** `rembak` がリモート・サーバーからの確認を待つタイムアウト期間を分単位で設定します。値が指定されないと、デフォルト・タイムアウトの 90 秒が設定されます。このデフォルトは、タイムアウトが 0 または負の値のときも使用されます。
- u UserName@HostName** `HostName` コンピューターから実行要求された `UserName` の印刷ジョブを取り消します。  
注: キューイング・システムは、マルチバイトのホスト名をサポートしません。
- X** リモート・サーバーが AIX 以外のマシンであっても、`rembak` コマンドがリモート・サーバーに `-o Option` を送信することを指定します。リモートが AIX 以外のマシンである場合、`Option` は `-o` フラグなしで送信されます。したがって、`-o -abc` は `-abc` として送信されます。  
リモート・キュー上で `-X` フラグを使用するには、特定のキューに対する以下の行を `/etc/qconfig` ファイルに含めなければなりません。  
`backend = /usr/lib/lpd/rembak -X`  
キュー上で `-X` を指定すると、`qprt`、`lpr` などのキュー・コマンドが機能しない場合があります。`enq` コマンドを使用してください。
- x** ジョブ要求を取り消します。`-# JobNumber` フラグまたは `-u UserName` フラグを使用して要求を取り消します。

## 例

1. AIX バージョン 2 オペレーティング・システムを使用する RT である、リモート・サーバー `olive` 上のキュー `popeye` にある、ファイル `spinach`、`asparagus`、および `broccoli` を印刷するには、次のように入力します。

```
rembak -S olive -P popeye -N /usr/lib/lpd/aixv2short spinach asparagus broccoli
```

2. キュー popeye に関して詳細状況要求を olive に出すには、次のように入力します。

```
rembak -S olive -P popeye -N /usr/lib/lpd/aixv2short -L
```

3. バージョン 3 のコンピューター bluto からユーザー sweetpea が実行要求したりモート・サーバー上のジョブ番号 23 を取り消すには、以下のように入力します。

```
rembak -S olive -P popeye -N /usr/lib/lpd/aixv2short -x -#23 -u sweetpea@bluto
```

## ファイル

`/usr/lib/lpd/rembak`  
`/etc/hosts.lpd`  
`/etc/hosts.equiv`

**rembak** コマンドが入っています。  
印刷要求を実行できるホスト名が入っています。  
印刷要求を実行できるホスト名が入っています。

## 関連情報

**cancel** コマンド、**chque** コマンド、**chqueuedev** コマンド、**disable** コマンド、**enable** コマンド、**enq** コマンド、**lp** コマンド、**lpd** コマンド、**lpq** コマンド、**lpr** コマンド、**lprm** コマンド、**lpstat** コマンド、**lsallq** コマンド、**lsallqdev** コマンド、**lsque** コマンド、**lsqueuedev** コマンド、**mkque** コマンド、**mkqueuedev** コマンド、**qadm** コマンド、**qcan** コマンド、**qchk** コマンド、**qdaemon** コマンド、**qpri** コマンド、**qprt** コマンド、**qstatus** コマンド、**rmque** コマンド、**rmqueuedev** コマンド。

`/etc/qconfig` ファイル。

プリンターおよび印刷 の 印刷の管理 のセクション。

プリンターおよび印刷 の 印刷スプーラー のセクション。

プリンターおよび印刷 の リモート印刷 のセクション。

---

## remove コマンド

### 目的

`var/adm/acct/sum` および `var/adm/acct/nite` サブディレクトリーからファイルを削除します。

### 構文

`/usr/sbin/acct/remove`

### 説明

**remove** コマンドは、`/var/adm/acct/sum(x)/wtmp*`、`/var/adm/acct/sum(x)/pacct*`、および `/var/adm/acct/nite(x)/lock*` ファイルをすべて削除します。**remove** コマンドは、**cron** デーモンを使用してスケジューリングしておかなければなりません。また、**remove** コマンドは毎晩ではなく、各アカウント期間の終わりに実行する必要があります。

### セキュリティ

アクセス制御: このコマンドは、**adm** グループのメンバーのみに実行 (x) アクセス権限を与えます。

## ファイル

|                                  |                                                    |
|----------------------------------|----------------------------------------------------|
| <code>/usr/sbin/acct</code>      | アカウント・コマンドへのパス。                                    |
| <code>/var/adm/acct/nite</code>  | アカウント・データ・ファイルが入っています。                             |
| <code>/var/adm/acct/nitex</code> | 8 文字を超えるユーザー名が使用されている場合、アカウント・データ・ファイルが入ります。       |
| <code>/var/adm/acct/sum</code>   | 日次アカウント・レコードの累積ディレクトリー。                            |
| <code>/var/adm/acct/sumx</code>  | 8 文字を超える長いユーザー名が使用された場合の日次アカウント・レコードのための累積ディレクトリー。 |

## 関連情報

アカウント・システム、日次および月次のレポートの準備、およびアカウント・ファイルについての詳細は、オペレーティング・システムおよびデバイスのマネージのシステム・アカウントのセクションを参照してください。

オペレーティング・システムおよびデバイスのマネージのアカウント・サブシステムの設定のセクションで、アカウント・システムの確立のための必要なステップが説明されています。

---

## removevsd コマンド

### 目的

仮想共用ディスクのセットを除去します。

### 構文

```
removevsd  {-v vsd_names | -a} [-f]
```

### 説明

仮想共用ディスクに関連付けられた論理ボリュームを除去するには、このコマンドを使用します。ボリューム・グループはこのコマンドでは除去されません。

仮想共用ディスクがシステム区画の任意のノードに構成されている場合、**-f** フラグが指定されていない限り、このコマンドは失敗します。

このコマンドは、System Management Interface Tool (SMIT) を使用して実行できます。SMIT を使用するには、以下のように入力します。

```
smit delete_vsd
```

さらに、**Remove a Virtual Shared Disk** オプションを選択します。

### フラグ

- v** 仮想共用ディスクの名前またはこのコマンドにより除去される名前を指定します。
- a** コマンドが、RSCT ピア・ドメイン内の仮想共用ディスクをすべて除去するように指定します。
- f** 仮想共用ディスクを構成解除し、除去するようシステムを強制実行します。**-f** が指定されず、除去される仮想共用ディスクのいくつかは構成されている場合は、このコマンドは失敗します。

## パラメーター

`vsd_name` 仮想共用ディスクを指定します。仮想共用ディスクが停止状態にない場合は、エラー・メッセージが出されます。

## セキュリティ

このコマンドを実行するには、**root** 権限が必要です。

## 制約事項

このコマンドはピア・ドメイン内のオンラインのノードから発行する必要があります。オンラインのピア・ドメインに移動するには **starttrpdomain** コマンドを使用します。既存のピア・ドメイン内で、オンラインの特定ノードに移動するには **starttrpnode** コマンドを使用します。RSCT ピア・ドメインの作成と管理の詳細については、「*RSCT Administration Guide*」を参照してください。

## 例

システムまたはシステム区画内の定義済みの仮想共用ディスクを、すべて構成解除し除去するには、次のように入力します。

```
removevsd -a -f
```

## 位置

`/opt/rsct/vsd/bin/removevsd`

## 関連情報

コマンド: **createvsd**

---

## renice コマンド

### 目的

実行中のプロセスの優先順位を変更します。

### 構文

```
renice [ -n Increment ] [ -g | -p | -u ] ID ...
```

### 説明

**renice** コマンドは、実行中の 1 つまたは複数のプロセスの `nice` の値を変更します。`nice` の値とは、プロセスのシステム・スケジューリング優先順位を表す 10 進値です。デフォルトでは、対象となるプロセスをプロセス ID で指定します。プロセス・グループを指定すると、要求はそのプロセス・グループ内のすべてのプロセスに適用されます。

`nice` の値は、インプリメンテーションに依存した方法で決定されます。要求した増分によって、実行されたユーティリティの `nice` の値が、インストールに依存する上下限を超える場合は、超過した値に対する制限値が適用されます。

ユーザーに `root` ユーザー権限がなければ、再設定できるのは自分自身のプロセスの優先順位のみで、0 から 20 までの範囲内で、20 を最下位の優先順位として、優先順位を大きくすることしかできません。`root`

ユーザー権限があれば、任意のプロセスの優先順位を変更し、優先順位を -20 から 20 までの範囲内の値に設定できます。指定した *Increment* により、以下のようにプロセスの順位が変更されます。

- 1 から 20        基本優先順位より遅い指定プロセスを実行します。
- 0                指定プロセスの優先順位を基本スケジューリング優先順位に設定します。
- 20 から -1     基本優先順位より速い指定プロセスを実行します。

**renice** コマンドは、これらの値をカーネルが実際に使用する値にマップします。

注:

1. root ユーザー権限がなければ、(最初に優先順位を小さくしたとしても) プロセスの *nice* の値を大きくすることはできません。
2. **renice** コマンドを使用して、一定の優先順位で実行するようにプロセスを変更することはできません。これを行うには、**setpriority** システム・コールを使用します。

## フラグ

- g                すべての ID を無符号の 10 進整数によるプロセス・グループ ID として解釈します。
- n *Increment*    プロセスの *nice* の値に追加する数値を指定します。*Increment* の値には、-20 から 20 までの 10 進整数しか入れることができません。正の増分値を指定すると、*nice* の値が低くなります。負の増分値を指定する場合は、該当する特権が必要になり、*nice* の値が高くなります。
- p                すべての ID を無符号整数によるプロセス・グループ ID として解釈します。-p フラグは、ほかのフラグを指定しない場合のデフォルトです。
- u                すべての ID をユーザー名または数値によるユーザー ID として解釈します。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0        正常終了。
- >0      エラーが発生しました。

## 例

1. 2 つのプロセス ID、987 と 32 のスケジューリング優先順位が低くなるように、システムのスケジューリング優先順位を変更するには、以下のように入力します。

```
renice -n 5 -p 987 32
```

2. 2 つのグループ ID、324 と 76 のスケジューリング優先順位が高くなるように、システムのスケジューリング優先順位を変更するには (ユーザーがこの操作を実行できる特権を持っている場合)、以下のように入力します。

```
renice -n -4 -g 324 76
```

3. 数値のユーザー ID 8 とユーザー ID *sas* のスケジューリング優先順位が低くなるように、システムのスケジューリング優先順位を変更するには、以下のように入力します。

```
renice -n 4 -u 8 sas
```

## ファイル

**/usr/sbin/renice**                **renice** コマンドが入っています。

## 関連情報

**nice** コマンド。

**getpriority** システム・コール、 **setpriority** システム・コール。

オペレーティング・システムおよびデバイスのマネージのコマンドのセクションで、コマンドの処理方法について説明しています。

オペレーティング・システムおよびデバイスのマネージのプロセスのセクションで、プロセスおよび入出力ダイレクトについて説明しています。

オペレーティング・システムおよびデバイスのマネージのシェルのセクションで、シェル、さまざまなシェルのタイプ、およびコマンドの解釈の仕方にシェルがどのように影響するかについて説明しています。

パフォーマンス・マネージメントの マイクロプロセッサの競合の制御のセクション。

---

## reorgvg コマンド

### 目的

ボリューム・グループに割り当てられた物理区画を再編成します。

### 構文

```
reorgvg [ -i ] VolumeGroup [ LogicalVolume ...]
```

### 説明

**注意:** このコマンドは、ボリューム・グループが並行モードでオンに変更されている場合は、許可されません。

**reorgvg** コマンドは、各論理ボリュームの割り当て特性にしたがって、*VolumeGroup* 内で割り当て済み物理区画の配置を再編成します。特定の論理ボリュームを再編成するには、*LogicalVolume* パラメーターを使用します。*LogicalVolume* パラメーター・リストの最初の論理ボリューム名に最上位の優先順位を、パラメーター・リストの最後の論理ボリュームに最下位の優先順位を与えます。**reorgvg** コマンドを使用するためには、ボリューム・グループをオンに変更しなければならず、また空き区画がなければなりません。

再編成を有効にするためには、**chlv -r** コマンドを使用して各論理ボリュームの再配置可能フラグを **y** に設定しなければなりません。それ以外の場合は、論理ボリュームが無視されます。

**注:**

1. **reorgvg** コマンドは、ストライプ化された論理ボリュームに割り当てられた物理区画の位置を認識しません。
2. **reorgvg** コマンドを正常に実行するためには、指定したボリューム・グループ上に最低 1 つの空き物理区画 (PP) がなければなりません。ミラーリングされた論理ボリュームの場合、**reorgvg** コマンドが実行時に論理ボリュームの正確性を保守するには、物理ボリューム (PV) につき 1 つの空き PP が必要です。そうしないと、**reorgvg** コマンドが依然として稼働し、論理区画の両方のコピーをその実行時に同じディスクに移動します。

3. このコマンドを使用するには、root ユーザー権を持っているか、または **system** グループのメンバーでなければなりません。
4. ボリューム・グループ名を持ち、ほかに引数を持たない **reorgvg** コマンドを入力すると、ボリューム・グループ全体が再編成されます。
5. **reorgvg** コマンドは、スナップショット・ボリューム・グループまたは、スナップショット・ボリューム・グループを持つボリューム・グループには使用できません。

Web-based System Manager (wsm) の ボリューム アプリケーションを使用してボリューム特性を変更できます。

このコマンドは、System Management Interface Tool (SMIT) の **smit reorgvg** 高速パスを使用して実行することもできます。

## フラグ

- i 標準入力から読み取る物理ボリュームを指定します。指定された物理ボリューム上の区画のみが編成されます。

## 例

1. ボリューム・グループ **vg02** 上の論理ボリューム **lv03**、**lv04**、および **lv07** を再編成するには、以下のように入力します。

```
reorgvg vg02 lv03 lv04 lv07
```

リストされた論理ボリュームのみが **vg02** 上で再編成されます。

2. 物理ボリューム **hdisk4** および **hdisk6** 上に配置された区画のうち、論理ボリューム **lv203** および **lv205** に属する区画のみを再編成するには、次のように入力します。

```
echo "hdisk4 hdisk6" | reorgvg -i vg02 lv203 lv205
```

ボリューム・グループ **vg02** の物理ボリューム **hdisk4** および **hdisk6** 上に配置された区画のうち、論理ボリューム **lv203** および **lv205** に属する区画が再編成されます。

## ファイル

|                          |                                   |
|--------------------------|-----------------------------------|
| <b>/usr/sbin/reorgvg</b> | <b>reorgvg</b> コマンドが入っているディレクトリー。 |
| <b>/tmp</b>              | コマンドの実行中に一時ファイルが格納されるディレクトリー。     |

## 関連情報

**chlv** コマンド、 **lslv** コマンド、 **mklv** コマンド。

オペレーティング・システムおよびデバイスのマネージの論理ボリューム・ストレージのセクション。

Web-based System Manager のインストールについては、 *AIX 5L Version 5.3 Web-based System Manager 管理ガイド* の第 2 章 インストールおよびシステム要件 のセクションを参照してください。

オペレーティング・システムおよびデバイスのマネージのSystem Management Interface Toolのセクション。

パフォーマンス・マネージメントの 論理ボリュームおよびディスク I/O のパフォーマンスのセクション。



---

## repl コマンド

### 目的

メッセージに応答します。

### 構文

```
repl [ +Folder ] [ -draftfolder +Folder | -nodraftfolder ] [ Message ] [ -draftmessage Message ]  
[ -annotate [ -noinplace | -inplace ] | -noannotate ] [ -cc Names... ] [ -nocc Names... ]  
[ -query | -noquery ] [ -fcc +Folder ] [ -form FormFile ] [ -editor Editor | -noedit ]  
[ -format | -noformat ] [ -filter File ] [ -width Number ] [ -whatnowproc Program |  
-nowhatnowproc ]
```

### 説明

**repl** コマンドは、ユーザーがメッセージに対する応答を作成できるように、インターフェースを始動させます。デフォルトでは、このコマンドは現行フォルダー内の現行メッセージの応答のドラフトを作成します。**-draftfolder** フラグを指定しないか、**\$HOME/mh\_profile** ファイル内の **Draft-Folder:** エントリーが定義されていない場合は、**repl** コマンドは MH ディレクトリー内で **draft** ファイルを検索します。フォルダーを指定すると、そのフォルダーが現行フォルダーになります。

**repl** コマンドを入力すると、システムは、ドラフト内に **To:**、**cc:**、および **In-Reply-To:** フィールドを置き、応答テキストの入力を求めるプロンプトを表示します。エディターを終了するには、**Ctrl-D** を押します。エディターが終了すると、**repl** コマンドは MH の **whatnow** コマンドを始動します。What now? プロンプトが表示されたときに **Enter** キーを押すと、使用可能な **whatnow** サブコマンドのリストを表示することができます。これらのサブコマンドを使用すると、応答の再編集、リスト、送信、または **repl** コマンド処理の終了を実行できます。

**注:** 送信時にメッセージが識別できるように、メッセージのヘッダーと本文の間に、ハイフンだけの行およびブランク行を 1 行入れておく必要があります。

**repl** コマンドは、**/etc/mh/replcomps** ファイル内の定義を使用して、応答メッセージをフォーマットします。MH ディレクトリー内で **replcomps** ファイルを作成したり、または **-form** フラグを使用して、代替応答フォーマットを定義することができます。オリジナル・メッセージのコピーをそのまま応答メッセージ内に残すには、**-filter** フラグを使用します。

元のメッセージに再配布情報に関する注釈を付けるには、**-annotate** フラグを使用します。このフラグは、オリジナル・メッセージに注釈として **Resent:** フィールドおよび現行日時を付けます。**repl** コマンド処理を終了する前に応答を送信した場合のみ、メッセージに注釈が付けられます。

### フラグ

|                             |                                                                                                                                              |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-annotate</b>            | 応答中のメッセージに注釈として応答日時を付けます。 <b>-inplace</b> フラグを使用すると、注釈付きメッセージへのリンクを保存することができます。                                                              |
| <b>-cc Names</b>            | 応答の <b>cc:</b> フィールドにリストされるユーザーを指定します。 <b>Names</b> には、 <b>all</b> 、 <b>to</b> 、 <b>cc</b> 、および <b>me</b> を指定できます。デフォルトは、 <b>-cc all</b> です。 |
| <b>-draftfolder +Folder</b> | 指定されたフォルダー内にドラフト・メッセージを入れます。 <b>+Folder</b> を指定しないと、 <b>Current-Folder</b> が使用されます。                                                          |

|                                     |                                                                                                                                                                                                          |
|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-draftmessage</b> <i>Message</i> | ドラフト・メッセージを指定します。 <b>-draftmessage</b> フラグを付けずに <b>-draftfolder</b> を指定すると、デフォルト・メッセージは新規となります。 <b>-draftfolder</b> フラグを付けずにこのフラグを指定すると、システムはデフォルト・ファイル <i>UserMhDirectory/draft</i> 内でドラフト・メッセージを作成します。 |
| <b>-editor</b> <i>Editor</i>        | 応答を構成する最初のエディターを識別します。 <b>-editor</b> フラグを指定しないと、 <b>comp</b> コマンドは、 <b>\$HOME/.mh_profile</b> ファイル内の Editor: エントリーで指定されたデフォルト・エディターを選択します。                                                              |
| <b>-fcc</b> <i>+Folder</i>          | 指定されたフォルダー内に応答のファイル・コピーを入れます。このフラグを指定しなければ、 <b>repl</b> コマンドはファイル・コピーを生成しません。                                                                                                                            |
| <b>-filter</b> <i>File</i>          | 応答されるメッセージを再フォーマットして、再フォーマットされたメッセージを応答文中に入れます。 <i>File</i> 変数を指定するときはこのフラグも指定しなければなりません。 <b>-filter</b> フラグは、 <b>mhl</b> コマンドで使えるフォーマット・ファイルを使用します。                                                     |
| <b>+Folder</b>                      | 応答対象となるメッセージが入っているフォルダーを識別します。フォルダーを指定しなければ、Current-Folder が使用されます。                                                                                                                                      |
| <b>-form</b> <i>FormFile</i>        | 応答フォーマットを指定します。 <b>repl</b> コマンドは、指定されたフォーマット・ファイル内の各行をフォーマット文字列として処理します。                                                                                                                                |
| <b>-format</b>                      | To:、cc:、および Bcc: フィールドから重複アドレスを除去し、 <b>-width</b> フラグで指定された桁数を使用してこれらのフィールドを標準化します。 <b>-format</b> フラグは、デフォルトとしてインターネット・フォーマットを使用するかどうかを示します。                                                            |
| <b>-help</b>                        | コマンド構文、使用可能なスイッチ (トグル)、およびバージョン情報をリストします。                                                                                                                                                                |
| <b>-inplace</b>                     | <b>注:</b> MH の場合は、このフラグ名を省略することはできません。注釈付けされたメッセージとのリンクを保存するために、強制的に同じ場所で注釈付けが行われるようにします。                                                                                                                |
| <i>Message</i>                      | メッセージを指定します。応答対象のメッセージとメッセージ・ドラフトの両方を指定する場合は、 <b>-draftmessage</b> フラグを使用しなければなりません。以下を使用してメッセージを定義します。                                                                                                  |
|                                     | <i>Number</i> メッセージの数。                                                                                                                                                                                   |
|                                     | <b>cur</b> または <b>.</b> (ピリオド)<br>現行メッセージ。デフォルトの応答メッセージ。                                                                                                                                                 |
|                                     | <b>first</b> フォルダー内の最初のメッセージ。                                                                                                                                                                            |
|                                     | <b>last</b> フォルダー内の最後のメッセージ。                                                                                                                                                                             |
|                                     | <b>new</b> 新たに作成したメッセージ。デフォルトのドラフト・メッセージは <b>new</b> です。                                                                                                                                                 |
|                                     | <b>next</b> 現行メッセージの次のメッセージ。                                                                                                                                                                             |
|                                     | <b>prev</b> 現行メッセージの前のメッセージ。                                                                                                                                                                             |
| <b>-noannotate</b>                  | 注釈を付けません。このフラグがデフォルトです。                                                                                                                                                                                  |
| <b>-nocc</b> <i>Names</i>           | 応答の cc: フィールド内にリストされないユーザーを指定できるようにします。 <i>Names</i> には <b>all</b> 、 <b>to</b> 、 <b>cc</b> 、および <b>me</b> を指定できます。                                                                                       |
| <b>-nodraftfolder</b>               | <i>UserMhDirectory/draft</i> 内にドラフトを配置します。                                                                                                                                                               |
| <b>-noedit</b>                      | 初期編集を抑制します。                                                                                                                                                                                              |
| <b>-noformat</b>                    | To:、cc:、および Bcc: フィールドから重複アドレスが除去されたり、これらのフィールドが標準化されないようにします。                                                                                                                                           |
| <b>-noinplace</b>                   | 定位置で注釈付けされないようにします。このフラグがデフォルトです。                                                                                                                                                                        |
| <b>-noquery</b>                     | To: および cc: フィールドを自動的に構築します。このフラグがデフォルトです。                                                                                                                                                               |
| <b>-nowhatnowproc</b>               | <b>repl</b> コマンドが対話式で処理されないようにします。このフラグを付けると編集できなくなります。                                                                                                                                                  |

|                                    |                                                                                   |
|------------------------------------|-----------------------------------------------------------------------------------|
| <b>-query</b>                      | To: および cc: フィールドに各アドレスを含めてもよいかどうかユーザーに照会します。                                     |
| <b>-whatnowproc</b> <i>Program</i> | ユーザーに応答タスクについて指示を与えるプログラムとして、指定のコマンド・ストリングを始動させます。デフォルトは、 <b>whatnow</b> プログラムです。 |
| <b>-width</b> <i>Number</i>        | アドレス・フィールドの幅を設定します。デフォルトは 72 桁です。                                                 |

## プロファイル・エントリー

下記のエントリーは、*UserMhDirectory/mh\_profile* ファイルに入力します。

|                      |                                              |
|----------------------|----------------------------------------------|
| Alternate-Mailboxes: | メールボックスを指定します。                               |
| Current-Folder:      | デフォルトの現行フォルダーを設定します。                         |
| Draft-Folder:        | ドラフト用のデフォルト・フォルダーを設定します。                     |
| Editor:              | デフォルトのエディターを設定します。                           |
| fileproc:            | メッセージをリファイルするのに使用するプログラムを指定します。              |
| mh1proc:             | 応答を作成中のメッセージをフィルターにかけるときに使用するプログラムを指定します。    |
| Msg-Protect:         | 新規メッセージ・ファイルの保護レベルを設定します。                    |
| Path:                | ユーザーの MH ディレクトリーを指定します。                      |
| whatnowproc:         | What now? という質問プロンプトを表示するときに使用するプログラムを指定します。 |

## 例

1. 現行フォルダーの現行メッセージに回答するには、以下のように入力します。

```
repl
```

システムは以下のようなテキストで応答します。

```
To: patrick@venus
cc: tom@thomas
Subject: Re: Meeting on Monday
In-reply-to: (Your message of Thu, 21 Jul 88 13:39:34 CST.)
<8807211839.AA01868>
```

```
-----
```

これで回答を入力できます。回答の本文を入力し終わったら、Ctrl-D キー・シーケンスを押してエディターを終了させます。システムは以下のように応答します。

```
What now?
```

回答を送信するには、send を入力します。サブコマンドのリストを表示したければ、Enter キーを押します。この例では、現行フォルダー内の現行メッセージに回答を送信することになります。

2. inbox フォルダー内のメッセージ 4 に回答するには、以下のように入力します。

```
repl +inbox 4
```

システムは次のようなメッセージで応答します。

```
To: dawn@chaucer
cc: jay@venus
Subject: Re: Status Report
In-reply-to: (Your message of Thu, 21 Jul 88 13:39:34 CST.)
<8807211839.AA01868>
```

```
-----
```

これで応答を入力できます。応答の本文を入力し終わったら、Ctrl-D キー・シーケンスを押してエディターを終了させます。システムは以下のように応答します。

```
What now?
```

応答を送信するには、`send` を入力します。サブコマンドのリストを表示したければ、`Enter` キーを押します。

3. 現行フォルダー内の現行メッセージへの応答を追跡するには、以下のように **-annotate** フラグを使用して、応答中のメッセージに日付と時刻をコピーします。

```
repl -annotate
```

システムは次のようなメッセージで応答します。

```
To: patrick@venus
cc: tom@thomas
Subject: Re: Meeting on Friday
In-reply-to: (Your message of Mon, 17 Apr 89 13:39:34 CST.)
<8904171839.AA01868>
-----
```

これで応答を入力できます。応答の本文を入力し終わったら、Ctrl-D キー・シーケンスを押してエディターを終了させます。システムは以下のように応答します。

```
What now?
```

応答を送信するには、`send` を入力します。応答を送信せずにエディターを終了させると、注釈は付けられません。

## ファイル

**\$HOME/mh\_profile**

**/etc/mh/replcomps**

*UserMhDirectory/replcomps*

**/usr/bin/repl**

*UserMhDirectory/draft*

ユーザーの MH プロファイルを指定します。

MH のデフォルトの応答テンプレートが入っています。

ユーザーのデフォルトの応答フォーマットが入っています。

**repl** コマンドが入っています。

現行メッセージのドラフトが入っています。

## 関連情報

**anno** コマンド、**comp** コマンド、**dist** コマンド、**forw** コマンド、**mhl** コマンド、**send** コマンド、**whatnow** コマンド。

**mh\_alias** ファイル・フォーマット、**mh\_profile** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ のメール・アプリケーションのセクション。

---

## replacepv コマンド

### 目的

ボリューム・グループ内の物理ボリュームを別の物理ボリュームで置き換えます。

## 構文

**replacepv** [ **-f** ] { *SourcePhysicalVolume* | *SourcePhysicalVolumeID* } *DestinationPhysicalVolume*

**replacepv** [ **-R** ] *dir\_name* [ *DestinationPhysicalVolume* ]

## 説明

**replacepv** コマンドは、割り当て済み物理区画とそこに含まれるデータを、*SourcePhysicalVolume* から *DestinationPhysicalVolume* に置き換えます。指定されたソース物理ボリュームは、*DestinationPhysicalVolume* と同じにはできません。

注:

1. *DestinationPhysicalVolume* サイズは、少なくとも *SourcePhysicalVolume* のサイズでなければなりません。
2. **replacepv** コマンドは、期限切れの論理ボリュームに期限切れでない状態ミラーがない限り、*SourcePhysicalVolume* をその状態論理ボリュームに置き換えることはできません。
3. **replacepv** コマンドは、スナップショット・ボリューム・グループまたは、スナップショット・ボリューム・グループを持つボリューム・グループには使用できません。

新しい物理区画の割り当ては、置き換えられる物理区画を含んでいる論理ボリューム用に定義されたポリシーに従います。

## フラグ

- f** *SourcePhysicalVolume* を、指定された *DestinationPhysicalVolume* と置き換えるように強制します。ただし、*DestinationPhysicalVolume* が、入出力装置構成データベースの別のボリューム・グループ、またはアクティブなボリューム・グループの一部である場合は除きます。
- R** *dir\_name* **replacepv** が <ctrl-c> (システム・クラッシュ) によって割り込まれた場合、または規定数が失われた場合に、このコマンドをリカバリーします。**-R** フラグを使用するときは、**replacepv** の初期実行中に指定されたディレクトリー名を指定する必要があります。このフラグを使用すると、*DestinationPhysicalVolume* を変更することもできます。

## セキュリティ

アクセス制御: このコマンドを実行するには、root 権限を持っていないければなりません。

## 例

1. hdisk1 から hdisk6 に物理区画を置き換える場合は、次のように入力します。

```
replacepv hdisk1 hdisk6
```

## ファイル

- /usr/sbin** **replacepv** コマンドが常駐しているディレクトリー。
- /tmp** コマンドの実行中に一時ファイルが格納されるディレクトリー。

## 関連情報

**migratepv** コマンド。

オペレーティング・システムおよびデバイスのマネージの論理ボリューム・ストレージのセクション。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

オペレーティング・システムおよびデバイスのマネージのSystem Management Interface Toolのセクション。

*AIX 5L Version 5.3 Kernel Extensions and Device Support Programming Concepts*のシステム・ダンプ機能のセクション。

---

## repquota コマンド

### 目的

ファイルシステムのクォータを要約します。

### 構文

```
repquota [ -v ] [ -c ] [ -g ] [ -u ] [ -l ] { -a | FileSystem ... }
```

### 説明

**repquota** コマンドは、*FileSystem* パラメーターで指定されたファイルシステムのクォータとディスク使用状況の要約を出力します。ファイルシステムの代わりに **-a** フラグを指定すると、**repquota** コマンドは、**/etc/filesystems** ファイル内の使用可能で、クォータを持つすべてのファイルシステムについて要約を出力します。デフォルトでは、ユーザー・クォータとグループ・クォータの両方が出力されます。

ユーザーごと、またはグループごとに、**repquota** コマンドは、以下を出力します。

- 既存のユーザー・ファイルまたはグループ・ファイルの数
- ユーザーまたはグループが使用中のディスク・スペース量
- ユーザー・クォータまたはグループ・クォータ

### フラグ

- a** **/etc/filesystems** ファイルの使用可能でクォータを持つすべてのファイルシステムについてクォータを出力することを指定します。
- c** コマンドの出力をコロンで区切るフォーマットに変更します。
- g** グループのクォータのみを出力することを指定します。
- l** 長いユーザー名を、**repquota** レポートで印刷できるようにします。レポートのデフォルト動作では、名前は 9 文字で切り捨てられます。**-l** オプションが指定されている場合は、完全なユーザー名が使用されます。
- u** ユーザーのクォータのみを出力することを指定します。
- v** ファイルシステムごとに、クォータ要約の前にヘッダー行を出力します。

## セキュリティ

アクセス制御: root ユーザーのみがこのコマンドを実行できます。

## 例

/u ファイルシステムのユーザー・クォータの要約を出力するには、以下のように入力します。

```
repquota -u /u
```

システムは以下の情報を出力します。

| User     | used | Block limits |      |        | used | File limits |      |       |
|----------|------|--------------|------|--------|------|-------------|------|-------|
|          |      | soft         | hard | grace  |      | soft        | hard | grace |
| root --  | 3920 | 0            | 0    |        | 734  | 0           | 0    |       |
| davec +- | 28   | 8            | 30   | 3 days | 3    | 0           | 0    |       |
| keith -- | 48   | 0            | 0    |        | 7    | 0           | 0    |       |

davec の次の欄に出力されている + は、ユーザーが設定済みブロック制限を超えたことを表しています。2 番目の欄に + があった場合、ユーザーが設定済みファイル制限値を超えたことを表します。

## ファイル

|                         |                        |
|-------------------------|------------------------|
| <b>quota.user</b>       | ユーザー・クォータを指定します。       |
| <b>quota.group</b>      | グループ・クォータを指定します。       |
| <b>/etc/filesystems</b> | ファイルシステムの名前と位置が入っています。 |
| <b>/etc/group</b>       | 基本グループ属性が入っています。       |
| <b>/etc/passwd</b>      | ユーザーの名前と位置が入っています。     |

## 関連情報

**quota** コマンド、 **quotaon** および **quotaoff** コマンド、 **edquota** コマンド、 **quotacheck** コマンド。

セキュリティのディスク・クォータ・システムの概要のセクション。

セキュリティのディスク・クォータ・システムの設定のセクション。

---

## reset コマンド

### 目的

端末を初期化します。

### 構文

```
reset [ -e C ] [ -k C ] [ -i C ] [ - ] [ -s ] [ -n ] [ -l ] [ -Q ] [ -m [ Identifier ] [ TestBaudRate ]  
:Type ] ... [ Type ]
```

### 説明

**reset** コマンドは、**tset** コマンドへのリンクとなっています。**tset** コマンドを **reset** コマンドとして実行すると、**tset** コマンドは端末依存処理を実行する前に以下のアクションを取ります。

- Cooked モードおよび Echo モードをオンに設定します。
- cbreak モードおよび Raw モードをオフにします。
- 改行変換をオンにします。
- 特殊文字を読み取れる状態に復元します。



特殊文字が NULL または -1 として検出された場合には、デフォルト値にリセットされます。**tset** コマンドに付けられるフラグは、すべて **reset** コマンドでも使用できます。

**reset** コマンドは、プログラムが停止した後で端末が望ましくない状態になっているときに、最も有効です。シーケンス <LF>reset<LF> (ここで、<LF> は行送りに相当する Ctrl-J です) は、この状態では復帰が機能しない場合があるので、**reset** コマンドを正常に実行させるために必要になることがあります。<LF>reset<LF> シーケンスはエコーされないことがよくあります。

## フラグ

- 決定した端末の名前が標準出力に出力されます。これは、シェルによって取り込まれ、**TERM** 環境変数に入れられることになっています。
- e C すべての端末上で消去文字を C 変数で指定した文字に設定します。デフォルトは、端末上のバックスペース文字で、通常は ^ (セディーユ) です。文字 C を直接入力することも、^ (セディーユ) を使用して入力することもできます。
- l 端末初期化文字列の送信を抑制します。
- i C **-e** フラグと類似していますが、消去文字ではなく割り込み文字を使用します。C 変数はデフォルト ^C にセットされます。このオプションに ^ 文字を使用することもできます。
- k C **-e** フラグと類似していますが、消去文字ではなく行消去文字を使用します。C 変数のデフォルトは ^X です。**-k** を指定しないと、抹消文字はそのままです。このオプションに ^ 文字を使用することもできます。
- mIdentifierTestbaudRate:Type Identifier パラメーター で識別されるポート上で、通常使用される端末タイプ (Type パラメーター内の) を指定します。存在しない ID は、すべての ID に一致します。TestBaudRate パラメーターにボー・レートを、任意に指定することもできます。
- n Berkeley 4.3 tty ドライバーを搭載したシステム上で、この端末について新規 tty ドライバー・モードを初期化する必要があることを指定します。CRT の場合、CRTERASE モードおよび CRTKILL モードを設定するのは、ボー・レートが 1200 bps 以上である場合のみです。詳しくは、**tty** ファイルを参照してください。
- Q 「Erase set to」メッセージと「Kill set to」メッセージを抑制します。
- s 決定した端末の名前に基づいて、**TERM** 環境変数を初期化する **csch** コマンドのシーケンスを出力します。

## ファイル

/usr/share/lib/terminfo/?/\* 端末機能データベースが入っています。

## 関連情報

**csch** コマンド、**sh** コマンド、**stty** コマンド、**tset** コマンド。

**environ** ファイル、**terminfo** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ の TTY 端末デバイスの各種ディスプレイおよび端末のための **TERM** 値のセクション。

---

## resize コマンド

### 目的

**TERMCAP** 環境変数および端末の設定を現在のウィンドウ・サイズに設定します。

### 構文

```
resize [ -c | -u ] [ -s [ Rows Columns ] ]
```

### 説明

**resize** コマンド・ユーティリティーは、**TERM** 環境変数および **TERMCAP** 環境変数がコマンドの実行元の **xterm** ウィンドウの現行サイズを示すようにするためのシェル・コマンドを出力します。この出力を有効にするために、**resize** コマンドを、コマンド行の一部として評価する (通常はシェルの別名または関数によって行う) か、または後で読み取ることができるファイルにリダイレクトしなければなりません。C シェル (通常は **/bin/csh**) から、以下の別名をユーザーの **.cshrc** ファイルに定義することができます。

```
% alias rs 'set noglob; `eval resize`'
```

ウィンドウのサイズ変更後、以下のように入力します。

```
% rs
```

コマンド関数のないバージョンの Bourne シェル (通常は **/bin/sh**) を使用しているユーザーは、以下のように出力を一時ファイルに送信し、その後で **.** (ドット) コマンドを用いて、それを読み取る必要があります。

```
$ resize >/tmp/out  
$ ./tmp/out
```

### フラグ

- c** ユーザーの現行シェルが **/bin/csh** でなくても、C シェル・コマンドが生成されることを指定します。
- u** ユーザーの現行シェルが Bourne シェルでなくても、Bourne シェル・コマンドが生成されることを指定します。
- s [Rows Columns]** 特殊な **xterm** エスケープ・コードの代わりに、Sun コンソール・エスケープ・シーケンスを使用することを指示します。 **Rows** パラメーターおよび **Columns** パラメーターを指定すると、**resize** コマンドは、**xterm** ウィンドウにそのウィンドウのサイズの変更を求めます。ただし、ウィンドウ・マネージャーは、変更を許可しない場合があります。

注: 両方を指定する場合は、**-c** または **-u** は、**-s** の左に表示する必要があります。

### ファイル

**/etc/termcap** 基本 **termcap** エントリーの変更を提供します。

### 関連情報

**csh** コマンド、**tset** コマンド、**xterm** コマンド。

---

## restart-secdapclntd コマンド

### 目的

**restart-secdapclntd** スクリプトは、現在実行中の **secdapclntd** デーモン・プロセスを停止してから、再始動するために使用されます。

### 構文

```
/usr/sbin/restart-secdapclntd [ -C CacheSize ] [ -p NumOfThread ] [ -t CacheTimeOut ] [ -T HeartBeatIntv ] [ -o ldapTimeOut ]
```

### 説明

**restart-secdapclntd** スクリプトは、**secdapclntd** デーモンが実行中であればこれを停止してから、再始動します。**secdapclntd** デーモンが実行中でない場合は、単にこれを開始します。

### フラグ

デフォルトでは、**secdapclntd** デーモンは始動時に、**/etc/security/ldap/ldap.cfg** ファイルに指定された構成情報を読み取ります。**secdapclntd** プロセスの開始時に、コマンド行に以下のオプションが指定されている場合、コマンド行のオプションは **/etc/security/ldap/ldap.cfg** ファイルの値を上書きします。

|                         |                                                                                                                                                                          |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-C</b> CacheSize     | <b>secdapclntd</b> デーモンが使用する最大キャッシュ・エントリーを、エントリーの CacheSize 数に設定します。ユーザー・キャッシュの場合、有効範囲は 100 から 10,000 エントリーです。デフォルトは 1000 です。グループ・キャッシュ・エントリーは、ユーザー・キャッシュ・エントリーの 10% です。 |
| <b>-o</b> ldapTimeOut   | サーバーに対する LDAP クライアント要求のタイムアウト期間 (秒単位)。この値は、クライアントが LDAP サーバーからの応答を待つ時間を決定します。有効範囲は 0 から 3600 (1 時間) です。デフォルトは 60 秒です。タイムアウトを使用不可にしてクライアントを強制的に無期限に待たせるには、この値を 0 に設定します。  |
| <b>-p</b> NumOfThread   | <b>secdapclntd</b> デーモンが使用するスレッド数を <b>NumOfThread</b> スレッドに設定します。有効な範囲は 1 から 1000 です。デフォルトは 10 です。                                                                       |
| <b>-t</b> CacheTimeOut  | キャッシュを CacheTimeOut 秒で期限切れにします。有効な範囲は 60 から 3600 秒です。デフォルト値は 300 秒です。                                                                                                    |
| <b>-T</b> HeartBeatIntv | このクライアントと LDAP サーバー間のハートビートの時間間隔を設定します。有効な値は 60 から 3,600 秒です。デフォルト値は 300 です。                                                                                             |

### 例

1. **secdapclntd** デーモンを再始動するには、次のように入力します。

```
/usr/sbin/restart-secdapclntd
```

2. 30 スレッドを使用し、500 秒のキャッシュ・タイムアウト値を使用して、**secdapclntd** を再始動するには、次のように入力します。

```
/usr/sbin/restart-secdapclntd -p 30 -t 500
```

### ファイル

**/etc/security/ldap/ldap.cfg** **secdapclntd** デーモンがサーバーに接続するために必要な情報が入っています。

## 関連情報

secldapIntd デーモン

mksecldap、stop-secldapIntd、start-secldapIntd、ls-secldapIntd、および flush-secldapIntd コマンド。

/etc/security/ldap/ldap.cfg ファイル。

---

## restbase コマンド

### 目的

ブート・イメージから基本カスタマイズ情報を読み取り、その情報をシステムのブート・フェーズ 1 で使用されるデバイス構成データベース内に復元します。

### 構文

```
restbase [ -o File ] [ -d Path ] [ -v ]
```

### 説明

**restbase** コマンドは、ブート・ディスクから基本カスタマイズ情報を読み取り、指定されたデバイス構成データベース・ディレクトリーに入れます。デフォルトでは、基本情報は、ブート・ディスクから読み取られます。デバイス構成データベース・ディレクトリーを指定しなければ、**restbase** コマンドは、この情報を **/etc/objrepos** ディレクトリーに復元します。**-o** フラグを使用すると、ブート・ディスク以外に、基本カスタマイズ情報の読み取り元となるファイルを指定できます。

**注意:** **restbase** コマンドは、システム・ブートのフェーズ 1 でのみ実行するためのコマンドです。実行時環境でそれを実行しないでください。そうでないと、デバイス構成データベースが破壊されることがあります。

### フラグ

**-o File**      基本カスタマイズ・データが入っているファイルを指定します。  
**-d Path**      基本デバイス構成データベースが入っているディレクトリーを指定します。  
**-v**            詳細出力を標準出力に書き出させます。

### 例

1. 基本カスタマイズ情報を復元し、詳細出力を表示するには、以下のように入力します。

```
restbase -v
```

2. 基本カスタマイズ情報を代替デバイス・データベースに復元するには、以下のように入力します。

```
restbase -d /tmp/objrepos
```

### ファイル

**/usr/lib/objrepos/PdDv**      このシステムがサポートする既知のデバイス・タイプすべてに関するエントリーが入っています。

|                                                                     |                                                                       |
|---------------------------------------------------------------------|-----------------------------------------------------------------------|
| <code>/etc/objrepos/CuDv</code>                                     | このシステムで定義されたすべてのデバイス・インスタンスに関するエントリーが入っています。                          |
| <code>/etc/objrepos/CuAt</code><br><code>/etc/objrepos/CuDep</code> | カスタマイズされたデバイス特有の属性情報が入っています。<br>ほかのデバイス・インスタンスに依存するデバイス・インスタンスを記述します。 |
| <code>/etc/objrepos/CuDvDr</code>                                   | デバイス構成ライブラリー・ルーチンによる並行管理が必要な重要リソースに関する情報を保管します。                       |

## 関連情報

**bosboot** コマンド、**savebase** コマンド。

*AIX 5L Version 5.3 Kernel Extensions and Device Support Programming Concepts* の Device Configuration Subsystem: Programming Introduction のセクション。

*AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグのプログラマーのためのオブジェクト・データ・マネージャー (ODM) の概要* のセクション。

*AIX 5L Version 5.3 Kernel Extensions and Device Support Programming Concepts* の List of Device Configuration Commands のセクション。

## restore コマンド

### 目的

**backup** コマンドで作成されたアーカイブからファイルを取り出します。

### 構文

#### ファイル名でアーカイブしたファイルを復元する

```
restore -x [ d M n Q v q e ] [ -b Number ] [ -f Device ] [ -s SeekBackup ] [ -E { force | ignore | warn } ] [ File ... ]
```

#### ファイル名でアーカイブしたファイルをリストする

```
restore -T | -t [ a l n q v Q ] [ -b Number ] [ -f Device ] [ -s SeekBackup ]
```

#### ファイルシステムでアーカイブしたファイルを復元する

```
restore -r [ B n q v y ] [ -b Number ] [ -f Device ] [ -s SeekBackup ]
```

#### ファイルシステムでアーカイブしたファイルを復元する

```
restore -R [ B n v y ] [ -b Number ] [ -f Device ] [ -s SeekBackup ]
```

#### ファイルシステムでアーカイブしたファイルを復元する

```
restore -i [ h m n q v y ] [ -b Number ] [ -f Device ] [ -s SeekBackup ]
```

#### ファイルシステムでアーカイブしたファイルを復元する

```
restore -x [ B h n m q v y ] [ -b Number ] [ -f Device ] [ -s SeekBackup ] [ File ... ]
```

## 指定されたボリューム番号からファイル復元を開始する

```
restore -X Number [ -MdnqveQ ] [ -b Number ] [ -f Device ] [ -s Number ] [ -E { force | ignore | warn } ] [ File ... ]
```

## ファイルシステムでアーカイブしたファイルをリストする

```
restore -t | -T [ B a l n h q v y ] [ -b Number ] [ -f Device ] [ -s SeekBackup ] [ File ... ]
```

## ファイル名でアーカイブしたファイル属性を復元する

```
restore -Pstring [ B d qv Q ] [ bNumber ] [ s SeekNumber ] [-f Device ] [ File ... ]
```

## ファイルシステムでアーカイブしたファイル属性を復元する

```
restore -Pstring [ hqv ] [ b Number ] [ s SeekNumber ] [-f Device] [ File ... ]
```

## 説明

**restore** コマンドは、**backup** コマンドで作成されたアーカイブを読み取って、そこに格納されているファイルを取り出します。このアーカイブは、ファイル名フォーマットまたはファイルシステム・フォーマットになっています。アーカイブは、ディスク、ディスケット、テープに格納することができます。ファイルは、アーカイブしたときと同じ方法で復元しなければなりません。そのためには、アーカイブのフォーマットを知る必要があります。アーカイブのフォーマットは、**-T** フラグを使用したときに表示されるアーカイブ・ボリューム・ヘッダー情報を検査することにより判別することができます。**-x**、**-r**、**-T**、**-t** フラグを使用すると、**restore** コマンドは自動的にアーカイブ・フォーマットを判別します。

**注:** **restore** は、復元されているファイルをアクティブに分散させます。ファイルに、NULL が記入されていて、ブロック調整およびサイズ変更された区域がある場合、**restore** によって、それらのファイルシステム・ブロック用の物理スペースは割り当てられません。バイト単位でのファイルのサイズは変わりませんが、ファイルシステム内で実際に使われるスペースは、NULL 以外の区域専用になります。

個々のファイルは、**-x** フラグを使用し、ファイル名を指定することにより、ファイル名アーカイブまたはファイルシステム・アーカイブから復元することができます。ファイル名は、アーカイブ上に存在しているとおりに指定しなければなりません。**-i** フラグを使うと、ファイルをファイルシステム・アーカイブから対話フォーマットで復元することができます。**-T** フラグを使用すると、アーカイブ上のファイルの名前を標準出力に出力できます。

ユーザーは、アーカイブの内容を抽出するために、ファイルシステム・デバイスへの書き込みアクセス、または、復元許可を得ておく必要があります。

ディスケット・デバイス、**/dev/rfd0** は、**restore** コマンドのデフォルトのメディアです。標準入力から復元するには、**-f** フラグに **-** (ハイフン) を指定します。また、**/dev/rmt0-2** のように、デバイスの範囲を指定することもできます。

### 注:

- 複数のボリュームからなるアーカイブから復元する場合、**restore** コマンドはマウントされているボリュームを読み取り、次のボリュームの挿入を求めるプロンプトを表示し、ユーザーからの応答を待ちます。新しいボリュームを挿入したら、Enter キーを押してファイルの復元を続けます。
- backup** コマンドを使用して、アーカイブをデバイス・ブロック・サイズが 0 に設定されたテープ・デバイスに作成した場合、そのテープから復元するには、テープの作成時に使ったブロック・サイズを正確に覚えていなければなりません。
- 単一のテープに複数のアーカイブが存在する場合があります。複数のアーカイブをテープから復元する場合、**restore** コマンドは入力デバイスがオープン時に保持せず、かつクローズ時に巻き戻さないテー



ブ・デバイスであるものと期待します。**-B**、**-s**、**-X** フラグのいずれかを指定しない限り、復元時に巻き戻さないテープ・デバイスを使わないでください。テープ・デバイスの使用については、**rmt** スペシャル・ファイルを参照してください。

## ファイルシステム・アーカイブ

ファイルシステム・アーカイブは、ファイルのアーカイブに使用される方法なので、**i** ノード・アーカイブとも呼ばれます。ファイルシステム名を **backup** コマンドで指定すると、ファイルシステム内のファイルは、ファイルシステム内の構造とレイアウトに基づいてアーカイブされます。**restore** コマンドは、ファイルシステムの基礎構造を特に認識せずに、ファイルシステム・アーカイブのファイルを復元します。

ファイルシステム・アーカイブを復元するとき、**restore** コマンドはファイル **restoresymtable** を作成して使用します。このファイルは、現行ディレクトリー内に作成されます。**restore** コマンドでファイルシステムの増分復元を行うには、このファイルが必要です。

**注:** ファイルシステムの増分バックアップおよび増分復元の実行時には、**restoresymtable** ファイルを除去しないでください。

**-r** または **-R** フラグを使うと、*File* パラメーターは無視されます。

## ファイル名アーカイブ

アーカイブするファイルのリストを **backup** コマンドで指定すると、ファイル名アーカイブが作成されます。**restore** コマンドは、ファイルシステムの基礎構造を特に認識せずに、ファイル名アーカイブからファイルを復元します。**restore** コマンドを使用すると、アーカイブから取り出すファイルを指定するときにメタキャラクターを指定できるようにします。これにより、パターン・マッチングに基づいてアーカイブからファイルを取り出す機能が使用できるようになります。パターン・ファイル名は単一引用符で囲み、パターンは大カッコで囲まなくてはなりません。 .

## スパース・ファイルについて

オペレーティング・システム・ファイルシステム (ヌル長文字列が含まれている) を、他のファイルよりもさらに効率的に保管できます。つまり、ヌル文字列が割り当てブロック全体にまたがる場合、その全ブロックはディスクに保管されません。このようにして 1 つまたは複数のブロックが省略されたファイルは、スパース・ファイルと呼ばれます。存在しないブロックはホール (穴) とも呼ばれます。

**注:** スパース・ファイルは、圧縮ファイルと同じではありません。スパース・ファイルが読み取られると、それらは非スパース同等ファイルとまったく同じになります。

スパース・ファイルは、通常、データベース・アプリケーションによって作成されます。データベース・ファイルは、作成されるたびに、ヌルでフォーマット設定されます。これらのフラグメントは、割り当てられたヌルで充てんされるか、または割り当てられていないヌルで充てんされるかのいずれかです。

## フラグ

- a** **t** または **T** (またはその両方) のオプションと一緒に指定し、**-a** オプションは、アーカイブ内のファイルのリストを権限と共にリストします。
- B** アーカイブを標準入力から読み取るように指定します。一般に、**restore** コマンドは実際のメディアを検査してバックアップ・フォーマットを判別します。**|** (パイプ接続) を使うと、この検査は実行できません。したがって、アーカイブはファイルシステム・フォーマット、デバイスは標準入力 (**-f -**) と見なされます。



**-b** *Number*

名前によるバックアップには 512 バイト・ブロックの数を指定し、i ノードによるバックアップには 1024 バイト・ブロックの数を指定して単一出力操作で読み取ります。

**restore** コマンドがテープ・デバイスから読み取る場合、名前によるバックアップのデフォルトは 100、i ノードによるバックアップのデフォルトは 32 です。

読み取りサイズは、ブロック数にブロック・サイズを掛けたものです。**restore** コマンドがテープ・デバイスから読み取るデフォルトの読み取りサイズは、名前によるバックアップの場合は 51200 (100 \* 512)、i ノードによるバックアップの場合は 32768 (32 \* 1024) です。読み取りサイズはテープの物理ブロック・サイズの偶数倍でなければなりません。読み取りサイズがテープの物理ブロック・サイズの偶数倍でなく、固定ブロック・モード (ゼロ以外) になっていると、**restore** コマンドは *Number* の有効値を判別しようとします。判別できれば、**restore** コマンドは *Number* を新しい値に変更して、変更結果に関するメッセージが標準出力に書き出され、処理は続行します。*Number* の有効値が見つからなければ、**restore** コマンドは標準エラーにエラー・メッセージを書き込み、ゼロ以外のリターン・コードを戻して終了します。*Number* パラメータの値が大きくなるほど、テープ・デバイスからの物理転送量が大きくなります。

**restore** コマンドは、ディスクからデータを読み取る場合は **-b** フラグの値を常に無視します。つまり、コマンドは常にトラック全体を占めるクラスターごとに読み取ります。

**-d**

*File* パラメータがディレクトリーであれば、そのディレクトリー内のファイルすべてを復元することを示します。このフラグは、アーカイブがファイル名フォーマットのときにしか使用できません。

**-e**

パック・ファイルとアンパック・ファイルのいずれの場合にも、非スパース・ファイルは、**backup** コマンドの名前フォーマットでアーカイブされたとおりに非スパースとして復元されます。ファイルをアーカイブする前に、ファイルのスパース性と非スパース性を知る必要があります。それは、このフラグを使用可能にすると、スパース・ファイルが非スパースとして復元されるからです。

このフラグは、4K を超える大きさのヌルを含む非スパースとしてファイルを復元する場合にのみ使用可能にする必要があります。**restore** 中に **-e** フラグを指定した場合、すべての通常ファイルが正常に復元され、また非スパース・データベース・ファイルは非スパースとして正常に復元されます。

**-E**

**-E** オプションを指定すると、指定されたボリューム番号から抽出が開始されます。このオプションには以下の引数のいずれかが必要です。**-E** オプションを省略すると、**warn** がデフォルトです。

**force** ファイルの固定エクステント・サイズまたはスペース予約を保てない場合は、ファイルの復元操作は失敗します。

**ignore** エクステント属性を保持するにあたってのエラーは、すべて無視します。

**warn** ファイルのスペース予約または固定サイズを保てない場合は、警告を出しません。

**-f** *Device*

入力デバイスを指定します。 名前の付いたデバイスから入力を受け取るには、 *Device* 変数をパス名 (`/dev/rmt0` など) で指定します。入力を標準出力デバイスから受け取るには、`-` (マイナス符号) を指定します。 `-` (マイナス) 機能により、`dd` コマンドからの **restore** コマンド入力のパイプ接続が可能になります。

また、アーカイブ・デバイスの範囲を指定することもできます。範囲指定には以下のフォーマットを使わなければなりません。

```
/dev/deviceXXX-YYY
```

ここで、*XXX* および *YYY* は整数であり、*XXX* は、常に、*YYY* より小さくなければなりません。例えば、`/dev/rfd0-3` です。

指定する範囲内のすべてのデバイスは、同じタイプでなければなりません。例えば、`8mm` で 2.3 ギガバイトのテープのセット、または 1.44 メガバイトのディスクットのセットを使用できます。すべてのテープ・デバイスは、同じ物理テープ・ブロック・サイズに設定しなければなりません。

*Device* 変数に範囲を指定すると、**restore** コマンドは範囲内の 1 つのデバイスから次のデバイスに自動的に進みます。指定されたデバイスをすべて使い果たすと、**restore** コマンドは一時停止し、デバイスの範囲に新しいボリュームをマウントするように要求します。

**-h**

実際のディレクトリーだけを復元し、このディレクトリーに入っているファイルは復元しません。このフラグは、アーカイブがファイルシステム・フォーマットの場合にしか使用できません。 `-r` または `-R` フラグと一緒に使うと、このフラグは無視されます。

**-i** 選択したファイルをファイルシステム・アーカイブから対話フォーマットで復元できるようにします。 **-i** フラグ用のサブコマンドは、下記のとおりです。

**cd** *Directory*

現行ディレクトリーを、指定したディレクトリーに変更します。

**add** [*File*]

*File* パラメーターを抽出ファイルのリストに追加するように指定します。*File* がディレクトリーであれば、そのディレクトリーと、そこに入っているすべてのファイルが抽出リストに追加されます (**-h** フラグを使用しない場合)。*File* を指定しなければ、現行ディレクトリーが抽出リストに追加されます。

**delete** [*File*]

*File* パラメーターを抽出ファイルのリストから除去するように指定します。*File* がディレクトリーであれば、そのディレクトリーと、そこに入っているすべてのファイルが抽出リストから除去されます (**-h** フラグを使用しない場合)。

**ls** [*Directory*]

*Directory* パラメーター内に入っているディレクトリーとファイルを表示します。ディレクトリー名は、名前の後に / (スラッシュ) が表示されます。指定したディレクトリー内のファイルとディレクトリーのうち、抽出リストに入っているものは、名前の前に \* (アスタリスク) が表示されます。詳細モードがオンになっていると、ファイルとディレクトリーの *i* ノード番号も表示されます。*Directory* パラメーターを指定しなければ、現行ディレクトリーが使用されます。

**extract** 抽出リスト上のすべてのディレクトリーとファイルを復元します。

**pwd** 現行ディレクトリーの絶対パス名を表示します。

**verbose**

**ls** サブコマンドがファイルとディレクトリーの *i* ノード番号を表示するようにします。アーカイブから取り出されるときに、各ファイルの詳細情報も表示されます。

**setmodes**

抽出リストに追加するすべてのディレクトリーのオーナー、モード、時刻を設定します。

**quit** **restore** を即座に終了させます。抽出リスト上のファイルは復元されません。

**help** サブコマンドの要約を表示します。

**-l** **-t** および **-T** オプションとともに指定します。これが指定されると、タイム・スタンプ、ファイルの権限、ファイル・サイズ、所有者、およびグループを含む、詳細なファイルのリストを表示します。 **-l** オプションは、 **-a** オプションをオーバーライドします。

**-M** 復元したファイルのアクセスおよび修正時間を復元時間に設定します。復元されたファイルが **ar** コマンドによって作成したアーカイブである場合、すべてのメンバー・ヘッダー内の修正時間も復元時間に設定します。名前の付いたファイルを個々に復元し、 **-x** または **-X** フラグも指定されている場合にしか、 **-M** フラグを指定することはできません。 **-M** フラグが指定されていない場合、 **restore** コマンドはアクセス時間と修正時間をバックアップ・メディアに表示されているとおりに保守します。

**-M** フラグは、データが AIX 4.2 バックアップの *i* ノードによるフォーマットか、または名前によるフォーマットであるときに使用します。

- m** 復元されたファイル名を、アーカイブ上に存在するファイルの *i* ノード番号に変更します。これは、復元するファイル数が少なく、それらのファイルに別のファイル名を付けて復元したい場合に便利です。復元されたアーカイブ・メンバーの名前は *i* ノード番号に変更されるので、ディレクトリー階層とリンクはともに保存されません。ディレクトリーとハード・リンクは、正規ファイルとして復元されます。**-m** フラグは、アーカイブがファイルシステム・フォーマットのときに使用します。
- n** デフォルトでは、`restore` コマンドは **ACL**、**PCL**、あるいは名前付き拡張属性がアーカイブにあれば、復元します。**-n** オプションを指定すると、`restore` コマンドは、アーカイブ内に **ACL**、**PCL**、あるいは名前付き拡張属性があってもスキップオーバーし、復元しません。
- Pstring** ファイル属性のみを復元します。ファイルの内容は復元しません。指定されたファイルがターゲット・ディレクトリー・パスに存在しない場合、ファイルは作成されません。このフラグは、文字列パラメーターに指定されたフラグによって、ファイル属性を選択して復元します。文字列パラメーターには、次の文字を組み合わせたことができます。
- A**     すべての属性を復元する。
  - a**     ファイルの権限のみを復元する。
  - o**     ファイルの所有権のみを復元する。
  - t**     ファイルのタイム・スタンプのみを復元する。
  - c**     ファイルの **ACL** 属性のみを復元する。
- 注: `restore` コマンドに存在するオプションの中で、オプション **v**、**h**、**b**、**s**、**f**、**B**、**d**、および **q** が、**P** オプションに有効です。**P** オプションは、ファイル名とファイルシステムの両方のアーカイブに使用することができます。**File** 引数がシンボリック・リンクの場合は、シンボリック・リンクのメタデータではなく、ターゲット・ファイルのメタデータが変更されます。
- 注意: スーパーユーザーにより実行された場合、**-P** フラグの指定により、他のユーザーが所有するファイルの属性を上書きします。
- Q** 名前によるバックアップの場合、何らかのタイプのエラーが発生した時は、リカバリーしてアーカイブを継続することを試みるのではなく、コマンドが終了するよう指定します。
- q** 最初のボリュームの使用準備ができており、`restore` コマンドがプロンプト「`mount the volume and hit Enter`」を表示する必要がないことを指定します。アーカイブが複数のボリュームにまたがっている場合、`restore` コマンドは後続のボリュームの挿入を求めるプロンプトを表示します。
- r** ファイルシステム・アーカイブ内のすべてのファイルを復元します。**-r** フラグは、全レベル 0 バックアップの復元、またはレベル 0 バックアップが復元された後の増分バックアップの復元に対してのみ使用されます。`restore` は、`restoresymtable` ファイルを使って、増分復元間で相互に情報を渡します。このファイルは、最後の増分バックアップが復元されたら除去してください。**-r** フラグを使用すると、*File* パラメーターは無視されます。
- R** 複数のボリュームからなるファイルシステム・アーカイブのうち、特定のボリュームを要求します。**-R** フラグを使用すると、以前に中断された復元を再開することができます。**-R** フラグを使用すると、*File* パラメーターは無視されます。再開後は、`restore` コマンドの動作は **-r** フラグを指定した場合と同じになります。

**-s** *SeekBackup*

バックアップを、複数のバックアップ・テープ・アーカイブ上で検索して復元するように指定します。**-s** フラグは、アーカイブをテープ・デバイスに書き込む場合にしか適用できません。**-s** フラグを正しく使用するには、`/dev/rmt0.1` または `/dev/rmt0.5` などのクローズ時に巻き戻さず、かつオープン時に保存しないテープ・デバイスを指定しなければなりません。巻き戻しテープ・デバイスの使用時に **-s** フラグを指定すると、**restore** コマンドはエラー・メッセージを表示し、ゼロ以外のリターン・コードを戻して終了します。巻き戻さないテープ・デバイスの使用時に **-s** フラグを指定しないと、**-s 1** のデフォルト値が使用されます。*SeekBackup* パラメーターの値は、1 から 100 の範囲内の値でなければなりません。**-s** フラグの動作の関係で、クローズ時に巻き戻さず、オープン時に保存しないテープ・デバイスを使う必要があります。**-s** に指定する値は、テープ上のアーカイブ位置ではなく、テープの読み取り/書き込みヘッドの位置に相対の位置です。例えば、複数のバックアップが入っているテープ・アーカイブから第 1、第 2、第 4 のバックアップ復元するには、**-s** フラグのそれぞれの値を **-s 1**、**-s 1**、**-s 2** と指定します。

**-t**

バックアップ・アーカイブに関する情報を表示します。アーカイブがファイルシステム・フォーマットであれば、アーカイブ上で見つかったファイルのリストが標準出力に書き出されます。各ファイル名の前には、アーカイブ上に存在していたときのファイルの *i* ノード番号が付いています。表示されるファイル名は、バックアップされたファイルシステムのルート (*/*) ・ディレクトリーに相対です。*File* パラメーターを指定しなければ、アーカイブ上のすべてのファイルがリストされます。*File* パラメーターを使用すると、そのファイルのみがリストされます。*File* パラメーターにディレクトリーを指定すると、そのディレクトリーに入っているすべてのファイルがリストされます。アーカイブがファイル名フォーマットの場合、ボリューム・ヘッダー内の情報は標準エラーに書き込まれます。このフラグを使用して、アーカイブがファイル名フォーマットであるか、ファイルシステム・フォーマットであるかを判別することができます。

**-T**

バックアップ・アーカイブに関する情報を表示します。アーカイブがファイル名フォーマットの場合、ボリューム・ヘッダー内の情報は標準エラーに書き込まれ、アーカイブ上で見つかったファイルのリストは標準出力に書き込まれます。ファイル名アーカイブの場合、*File* パラメーターは無視されます。アーカイブがファイルシステム・フォーマットであれば、動作は **-t** フラグと同じになります。

**-v**

復元中に詳細情報を表示します。アーカイブがファイル名フォーマットである場合に、**-x** か **-T** フラグのいずれかが指定されると、アーカイブ上に存在していたときのファイルのサイズがバイト単位で表示されます。ディレクトリー、ブロック、またはキャラクター・デバイス・ファイルは、サイズ 0 でアーカイブされています。シンボリック・リンクは、そのサイズとともにリストされます。ハード・リンクは、アーカイブ方法を示すファイルのサイズとともにリストされます。アーカイブが読み取られると、これらのサイズの合計が表示されます。アーカイブがファイルシステム・フォーマットであれば、ディレクトリーと非ディレクトリーのアーカイブ・メンバーが区別されます。

-x

*File* パラメーターで指定したファイルを個別に復元します。*File* パラメーターを指定しなければ、すべてのアーカイブ・メンバーが復元されます。*File* パラメーターがディレクトリーであり、アーカイブがファイル名フォーマットであれば、ディレクトリーのみが復元されます。*File* パラメーターがディレクトリーであり、アーカイブがファイルシステム・フォーマットであれば、このディレクトリーに入っているすべてのファイルが復元されます。*File* パラメーターで指定するファイル名は、**restore -T** コマンドで表示される名前と同じでなければなりません。ファイルは、アーカイブ時と同じ名前で復元されます。相対パス名 (*.filename*) を使用してファイル名がアーカイブされた場合には、ファイルは現行ディレクトリーに相対的な位置に復元されます。アーカイブがファイルシステム・フォーマットであれば、ファイルは現行ディレクトリーに相対的な位置に復元されます。

**restore** コマンドは、必要なディレクトリーを自動的に作成します。このフラグを使用してファイルシステム・バックアップを復元すると、先頭ボリューム番号の入力を求めるプロンプトが表示されます。

**restore** コマンドを使用すると、アーカイブから取り出すファイルを指定するときに、シェル・スタイル・パターン・マッチング・メタキャラクターを指定できるようにします。マッチング・メタキャラクターに関する規則は、次のシェル・パス名 "globbing" で使用される規則と同じです。

#### \* (アスタリスク)

ゼロまたはそれ以上の文字と一致しますが、「.」(ピリオド) または「/」(スラッシュ) とは一致しません。

#### ? (疑問符)

任意の 1 文字と一致しますが、「.」(ピリオド) または「/」(スラッシュ) とは一致しません。

#### [ ] (大カッコ)

大カッコで囲んだ文字のうちの任意の 1 文字と一致します。大カッコの中にダッシュで区切られた文字の対が入っている場合、パターンは、辞書的には、現在ローカルであるそれらの 2 つの文字の間に入るどの文字とも一致します。さらに、大カッコの中の「.」(ピリオド) または「/」(スラッシュ) は、ファイル名の中の「.」(ピリオド) または「/」(スラッシュ) とは一致しません。

#### \ (バックスラッシュ)

その直後に続く文字と一致し、それがメタキャラクターとして解釈されないようにします。

-X *VolumeNumber*

複数のボリュームからなるファイル名バックアップのうち、指定したボリュームから復元を開始します。始動後は、**restore** コマンドの動作は **-x** フラグを指定した場合と同じになります。**-X** フラグはファイル名アーカイブにのみ適用されます。

-y

テープ・エラーが発生した場合に復元を続行します。通常、**restore** コマンドは処理を続行するかどうかを尋ねるプロンプトを表示します。どちらの場合も、読み取りバッファ内のすべてのデータはゼロに置き換えられます。**-y** フラグは、アーカイブがファイルシステム・フォーマットの場合にのみ適用されます。

-?

使用方法メッセージを表示します。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0 正常終了。
- >0 エラーが発生しました。



## 例

1. ディスケット・デバイス **/dev/rfd0** 上の、ファイル名アーカイブまたはファイルシステム・アーカイブ内のファイルの名前をリストするには、以下のように入力します。

```
restore -Tq
```

アーカイブは、デフォルトの復元デバイス **/dev/rfd0** から読み取られます。アーカイブに入っているすべてのファイルとディレクトリーの名前が表示されます。ファイルシステム・アーカイブの場合、ファイル名の前には、アーカイブ上に存在していたときのファイルの **i** ノード番号が付いています。**-q** フラグは **restore** コマンドに対して、第 1 のボリュームが使用可能であり、読み取る準備ができていることを指示します。したがって、第 1 のボリュームのマウントを求めるプロンプトは表示されません。

2. 特定のファイルを復元するには、以下のように入力します。

```
restore -xvqf myhome.bkup system.data
```

このコマンドは、ファイル **system.data** を、アーカイブ **myhome.bkup** から現行ディレクトリーに取り出します。この例にあるアーカイブは現行ディレクトリーにあります。ファイル名とディレクトリー名は、**-T** フラグの使用時に表示されるとおりに指定しなければなりません。**-v** フラグを指定すると、抽出中に詳細情報が表示されます。この例は、ファイル名アーカイブとファイルシステム・アーカイブの両方に該当します。

3. 特定のディレクトリーとそのディレクトリーの内容を、ファイル名アーカイブから復元するには、以下のように入力します。

```
restore -xdvqf /dev/rmt0 /home/mike/tools
```

**-x** フラグは **restore** コマンドに対して、ファイルをファイル名別に取り出すように指示します。**-d** フラグは **restore** コマンドに対して、**/home/mike/tools** ディレクトリーに入っているファイルとサブディレクトリーをすべて取り出すように指示します。ファイル名とディレクトリー名は、**-T** フラグの使用時に表示される通りに指定しなければなりません。ディレクトリーが存在しなければ、作成されません。

4. 特定のディレクトリーとそのディレクトリーの内容を、ファイルシステム・アーカイブから復元するには、以下のように入力します。

```
restore -xvqf /dev/rmt0 /home/mike/tools
```

このコマンドは、ファイルをファイル名別に取り出します。ファイル名とディレクトリー名は、**-T** フラグの使用時に表示されるとおりに指定しなければなりません。ディレクトリーが存在しなければ、作成されます。

5. ファイルシステム全体のアーカイブを復元するには、以下のように入力します。

```
restore -rvqf /dev/rmt0
```

このコマンドは、テープ・デバイス **/dev/rmt0** にアーカイブされたファイルシステム全体を、現行ディレクトリーに復元します。この例では、復元するファイルシステムのルート・ディレクトリーに現在いるものと想定しています。アーカイブが増分ファイルシステム・アーカイブのセットの一部であれば、アーカイブは、レベル 0 から始まり増加していくバックアップ・レベルで (0、1、2 など) 復元する必要があります。

6. 単一ボリュームの複数バックアップ・テープから第 5 および第 9 バックアップを復元するには、以下のように入力します。

```
restore -xvqs 5 -f/dev/rmt0.1  
restore -xvqs 4 -f/dev/rmt0.1
```



最初のコマンドは、`/dev/rmt0.1` で指定した複数バックアップ・テープ上の第 5 アーカイブから、すべてのファイルを取り出します。`.1` 指定子は、テープ・デバイスがオープン時に保存されず、クローズ時に巻き戻されないように指定します。`-s` フラグの動作の関係で、クローズ時に巻き戻さず、オープン時に保存しないテープ・デバイスを使う必要があります。第 2 のコマンドは、第 4 アーカイブ (テープ上のテープ・ヘッドの現行位置に相対) からすべてのファイルを取り出します。第 5 のアーカイブを復元後、テープの読み取り/書き込みヘッドはアーカイブを読み取る位置にあります。テープ上の第 9 のアーカイブを取り出すためには、`-s` フラグで値 4 を指定しなければなりません。これは、`-s` フラグがテープ上のアーカイブとの相対位置ではなく、テープ上の現行位置との相対位置だからです。第 9 のアーカイブは、テープ上の現行位置を基準にすると第 4 のアーカイブです。

- 10 本のテープからなる複数バックアップ・アーカイブのうち、第 6 のテープから始まる第 4 のバックアップを復元するには、第 6 のテープをテープ・ドライブに挿入して以下のように入力します。

```
restore -xcs 2 -f /dev/rmt0.1 /home/mike/manual/chap3
```

第 4 のバックアップが第 6 のテープ上の第 2 のバックアップであると仮定した場合、`-s 2` を指定するとテープ・ヘッドはこのテープ上の第 2 のバックアップの先頭は移動します。`restore` コマンドは、指定されたファイルをアーカイブから復元します。バックアップが後続のボリュームに継続しており、ファイルをまだ復元し終わっていないければ、`restore` コマンドはバックアップの終わりに達するまで、次のボリュームの挿入を求めるプロンプトを表示します。`-f` フラグは、巻き戻さず、かつ保存しないテープ・デバイス名を指定します。

**注:** `-s` フラグは、テープ・ドライブに挿入されたテープに対して相対であり、全 10 本のテープからなるアーカイブに対して相対でないバックアップ番号を指定します。

8. ストリーミング・テープ・デバイスのパフォーマンスをよくするには、以下のように入力して `dd` コマンドを `restore` コマンドにパイプ接続します。

```
dd if=/dev/rmt0 bs=64b | restore -xf- -b64
```

`dd` コマンドは、64 (512 バイト・ブロック) からなるブロック・サイズを使用して、テープからアーカイブを読み取り、標準出力に書き出します。`restore` コマンドは、64 (512 バイト・ブロック) からなるブロック・サイズを使用して、標準入力を読み取ります。`dd` コマンドがテープからアーカイブを読み取るために使用するブロック・サイズの値は、`backup` コマンドでテープを作成するときを使用したブロック・サイズの偶数倍でなければなりません。例えば、以下の `backup` コマンドを使用した場合に、この例で取り出されるアーカイブを作成することができます。

```
find /home -print | backup -ivqf/dev/rmt0 -b64
```

この例は、ファイル名フォーマットのアーカイブにのみ適用されます。アーカイブがファイルシステム・フォーマットの場合は、`restore` コマンドに `-B` フラグを指定する必要があります。

9. 9348 磁気テープ装置モデル 12 において `restore` コマンドのパフォーマンスをよくするには、以下のように入力してブロック・サイズを変更できます。

```
chdev -l DeviceName -a BlockSize=32k
```

10. 非スパース・データベース・ファイルを復元するには、次のように入力します。

```
restore -xef /dev/rmt0
```

11. アーカイブの前にスパースであったファイルをスパースとして復元するには、以下のように入力します。

```
restore -xf /dev/rmt0
```

12. ファイルの権限のみをアーカイブから復元するには、次のように入力します。

```
restore -Pa -vf /dev/rmt0
```

13. ファイルの ACL 属性のみをアーカイブから復元するには、次のように入力します。

```
restore -Pc -vf /dev/rmt0
```

14. ファイルの権限と目次を表示するには、次のように入力します。

```
restore -Ta -vf /dev/rmt0
```

15. タイム・スタンプとファイルの権限とともにファイル名アーカイブの目次を表示するには、次のように入力します。

```
restore -Tl -vf /dev/rmt0
```

16. タイム・スタンプとファイルの権限とともにファイルシステム・アーカイブの目次を表示するには、次のように入力します。

```
restore -tl -vf /dev/rmt0
```

## ファイル

`/dev/rfd0`

デフォルトの復元デバイスを指定します。

`/usr/sbin/restore`

**restore** コマンドが入っています。

## 関連情報

**ar** コマンド、**backup** コマンド、**mkfs** コマンド、**fsck** コマンド、**dd** コマンド。

**filesystems** ファイル、**backup** ファイル、**rmt** スペシャル・ファイル。

オペレーティング・システムおよびデバイスのマネージのバックアップ方式のセクション。

オペレーティング・システムおよびデバイスのマネージのファイルシステムのセクションは、ファイルシステムのタイプ、管理、構造、および保守について説明しています。

オペレーティング・システムおよびデバイスのマネージのSystem Management Interface Toolのセクション。

---

## restorevgfiles コマンド

### 目的

バックアップ・ソースからファイルを復元します。

### 構文

```
restorevgfiles [ -b blocks ] [ -f device ] [ -a ] [ -n ] [ -s ] [ -d path ] [ -D ] [ file_list ]
```

### 説明

**restorevgfiles** コマンドは、テープ、ファイル、CD-ROM、またはこれらのボリューム・グループ・バックアップ・ソースからファイルを復元します。**restorevgfiles** コマンドは、複数の CD、DVD、USB ディスク、またはテープなどのマルチボリューム・バックアップにも作用します。

**restorevgfiles** コマンド、または **listvgbackup-r** コマンドは、同一の操作を実行するので、交換可能であることを考慮する必要があります。**restorevgfiles** コマンドは自動的に **-r** フラグを適用します。**-r** フラグは冗長ですが、互換性を目的として保存され、指定した場合、通常にない振る舞いの原因にはなりません。**-r** フラグの完全な説明は、**listvgbackup** コマンドを参照してください。

## フラグ

|                         |                                                                                                                                                                                       |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-b</b> <i>blocks</i> | <i>blocks</i> パラメーターで定義して、1 回の入力操作で読み取る 512 バイト・ブロックの数を指定します。 <i>blocks</i> パラメーターが指定されていない場合は、読み取るブロックの数はデフォルトの 100 になります。                                                           |
| <b>-f</b> <i>device</i> | <i>device</i> パラメーターで定義して、バックアップ (ファイル、テープ、CD-ROM、またはその他のソース) を収めているデバイスのタイプを指定します。 <b>-f</b> を指定していない場合は、 <i>device</i> はデフォルトの <b>/dev/rmt0</b> になります。                              |
| <b>-a</b>               | <b>-b block</b> フラグで指定して、テープ・バックアップの物理ブロック・サイズを検査します。必要に応じて、バックアップを読み取るブロック・サイズを変更します。 <b>-a</b> フラグは、テープ・バックアップを使用する場合のみ有効です。                                                        |
| <b>-n</b>               | ACL、PCL、あるいは拡張属性を、復元しません。                                                                                                                                                             |
| <b>-s</b>               | バックアップ・ソースがユーザー・ボリューム・グループで、 <b>rootvg</b> でないことを指定します。                                                                                                                               |
| <b>-d</b> <i>path</i>   | <i>path</i> パラメーターで定義して、ファイルが復元される先のファイルのディレクトリー・パスを指定します。 <b>-d</b> パラメーターを使用しないと、現在の作業ディレクトリーが使用されます。現在の作業ディレクトリーが <b>root</b> の場合は、問題が起きます。 <b>root</b> の代わりに一時フォルダーへ書き込むことを推奨します。 |
| <b>-D</b>               | デバッグの出力を生成します。                                                                                                                                                                        |

## パラメーター

|                  |                                                                                                                                                                                        |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>file_list</i> | 復元されるファイルのリストを識別します。現行ディレクトリーに関連するファイルの絶対パスは、スペースで区切ったリストに指定する必要があります。その他ファイルが指定されないかぎり、指定されたディレクトリーの中のファイルがすべて復元されます。ディレクトリー内の全ファイルを復元している場合は、 <b>root</b> の代わりに一時フォルダーへ書き込むことを推奨します。 |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## 例

- /dev/cd1** で保存されたバックアップを読み取り、**/data/myfiles** ディレクトリーに全ファイルを保管するには、次を入力します。  

```
restorevgfiles -f /dev/cd1 -s -d /data/myfiles
```
- 一度に 20 個の 512 バイト・ブロックでデフォルト・デバイスから **vg** ユーザー・バックアップを読み取り、**/myapp/app.h** ファイルを現行ディレクトリーに復元するには、次を入力します。  

```
restorevgfiles -b 20 -s ./myapp/app.h
```
- /dev/cd1** で保管されたバックアップを読み取り、**/myapp/app.c** ファイルを **/data/testcode** ディレクトリーに復元するには、次を入力します。  

```
restorevgfiles -f /dev/cd1 -s -d /data/testcode ./myapp/app.c
```
- /dev/usbms0** に保存されているバックアップを読み取り、すべてのファイルを **/data/myfiles** ディレクトリーに復元するには、次のコマンドを入力します。  

```
restorevgfiles -f /dev/usbms0 -s -d /data/myfiles
```

## ファイル

**/usr/bin/restorevgfiles**

**restorevgfiles** コマンドが入っています。

## 関連情報

listvgbackup コマンド。

---

## restvg コマンド

### 目的

ユーザー・ボリューム・グループと、そのグループのすべてのコンテナとファイルを復元します。

### 構文

```
restvg [ -b Blocks ] [ -d FileName ] [ -f Device ] [ -l ] [ -q ] [ -r ] [ -s ] [ -n ] [ -P PPsize ] [ DiskName ... ]
```

### 説明

**restvg** コマンドは、**savevg** コマンドで作成したバックアップ・イメージに含まれる `/tmp/vgdata/vgname/vgname.data` ファイル (ここで *vgname* はボリューム・グループ名) に指定したとおりに、ユーザー・ボリューム・グループと、そのグループのすべてのコンテナとファイルを復元します。

**restvg** コマンドはユーザー・ボリューム・グループを復元します。**bosinstall** ルーチンは、ルート・ボリューム・グループ (**rootvg**) を再インストールします。**restvg** コマンドがバックアップ・イメージで **rootvg** ボリューム・グループを見つけると、**restvg** コマンドはエラーで終了します。

値 **yes** が `/tmp/vgdata/vgname/vgname.data` ファイルにある **logical\_volume\_policy** スタンザの **EXACT\_FIT** に指定されていると、**restvg** コマンドはマップ・ファイルを使用してそれぞれの論理ボリュームの物理区画の配置を保持します。ターゲット・ディスクは、`vgname.data` ファイルにある **source\_disk\_data** スタンザにそのとき指定されたソース・ディスクと同じサイズか、それよりも大きくなくてはなりません。

#### 注:

- バックアップ・イメージにあるファイルを調べたり、バックアップ・イメージからそれぞれのファイルを復元するには、**-T** または **-x** フラグをそれぞれ指定した **restore** コマンドを使用しなければなりません。(詳細は **restore** コマンドを参照してください。)
- ボリューム・グループ上で **varyonvg** コマンドを実行時、論理トラック・グループ (LTG) サイズはそのディスクの共通最大転送サイズに設定されます。

### フラグ

**-b** *Blocks*

1 回の入力操作で読み取る 512 バイト・ブロックの数を指定します。このパラメーターが指定されていない場合、**restore** コマンドは 100 のデフォルトを使用します。値が大きくなるほど、テープ・デバイスへの物理的な転送量は大きくなります。

|                           |                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>DiskName...</i>        | <b>vgname.data</b> ファイルにリストされているディスク・デバイスではなく、使用するディスク・デバイスの名前を指定します。ターゲット・ディスク・デバイスは空の物理ボリュームとして定義されなくてはなりません。つまり、物理ボリューム ID を含み、ボリューム・グループに属してはなりません。ターゲット・ディスク・デバイスが新規のものである場合は、 <b>mkdev</b> コマンドを使用してシステムに追加する必要があります。ターゲット・ディスク・デバイスがボリューム・グループのものである場合は、 <b>reducevg</b> コマンドを使用してボリューム・グループから除去しなくてはなりません。                                                                  |
| <b>-d</b> <i>FileName</i> | <b>-d</b> フラグはオプションのフラグであり、指定する場合は、続けてファイル名を指定する必要があります。このファイルは、復元されるバックアップ・イメージ内にある <b>vgname.data</b> ファイルの代わりとして使用されます。ファイル名は、相対パス名または絶対パス名として指定することができます。                                                                                                                                                                                                                       |
| <b>-f</b> <i>Device</i>   | バックアップ・メディアのデバイス名を指定します。デフォルトは <b>/dev/rmt0</b> です。                                                                                                                                                                                                                                                                                                                                 |
| <b>-l</b>                 | ボリューム・グループ・バックアップに関する便利な情報を表示します。                                                                                                                                                                                                                                                                                                                                                   |
|                           | このフラグは <b>-f device</b> フラグを必要とします。このフラグを指定すると、 <b>restvg</b> は、ボリューム・グループ、バックアップ作成日時、バックアップされたシステムからの <b>uname</b> 出力、推奨する保守およびテクノロジー・レベル、バックアップ・サイズ (メガバイト単位)、およびバックアップ縮小サイズ (メガバイト単位) といった情報を表示します。縮小サイズは、すべてのファイルシステム上のデータのサイズです。フルサイズは、各ファイルシステムの合計サイズ (未使用 + データ) です。 <b>-l</b> フラグは、バックアップされたボリューム・グループの論理ボリュームとファイルシステムの情報も表示し、これは " <b>lsvg -l vgname</b> " を実行したものに相当します。 |
| <b>-n</b>                 | 既存の MAP ファイルが無視されることを指定します。 <b>-n</b> フラグは、 <b>vgname.data</b> ファイルの <b>logical_volume_policy</b> スタンザ内の <b>EXACT_FIT</b> フィールド値を指定変更します。                                                                                                                                                                                                                                           |
| <b>-P</b> <i>PPsize</i>   | それぞれの物理区画のメガバイト数を指定します。メガバイト数を指定しないと、 <b>restvg</b> は、復元先の 1 番大きいディスクに合わせて <b>PPsize</b> にその最大値を使用します。この値が <b>vgname.data</b> ファイルに指定されたサイズと異なる場合、個々の論理ボリュームの区画数は、新しい <b>PPsize</b> に合わせて適切に変更されます。                                                                                                                                                                                 |
|                           | 適切なディスク・サイズよりも小さい <b>PPsize</b> が指定されている場合は、大きい方の <b>PPsize</b> が使用されます。                                                                                                                                                                                                                                                                                                            |
|                           | 適切なディスク・サイズよりも大きい <b>PPsize</b> が指定されている場合は、指定の大きい方の <b>PPsize</b> が使用されます。                                                                                                                                                                                                                                                                                                         |
| <b>-q</b>                 | ボリューム・グループ・イメージの復元前に通常のプロンプトが表示されないように指定します。このフラグを指定しないと、プロンプトがボリューム・グループ名とターゲット・ディスク・デバイス名を表示しません。                                                                                                                                                                                                                                                                                 |

-r

ボリューム・グループ構造のみを再作成します。これは、`restvg` が (指定されたバックアップ `FileName` または `Device` について)、ファイルまたはデータの復元を行わずに、バックアップから、ボリューム・グループ、論理ボリューム、およびファイルシステムを作成できるようにします。このフラグを使用すると便利なユーザーとは、データを復元するためにサード・パーティーのソフトウェアを使用し、単にすべての AIX 論理ボリューム構造が整っていることを必要とするユーザーです。

**注:** `-f Device` フラグまたは `-d FileName` フラグと一緒に使用します。この理由は、`restvg` は、必要なボリューム・グループの論理ボリューム構造の再作成に必要なすべての情報を、バックアップ・イメージまたは `vgname.data` ファイルから入手するためです。

-s

論理ボリュームをファイルシステムにおさまるようにできるだけ小さいサイズで作成するように指定します。このサイズは `vgname.data` ファイル (ここで、`vgname` はボリューム・グループの名前) にある `lv_data` スタンザの `LV_MIN_LPS` フィールドの値に指定します。

`-s` フラグは `vgname.data` ファイルにある `logical_volume_policy` スタンザの `SHRINK` フィールドと `EXACT_FIT` フィールドの値をオーバーライドします。`-s` フラグを指定すると、`SHRINK=yes` および `EXACT_FIT=no` の値を指定した場合と同じ結果を得ます。

## 例

1. `/dev/rmt1` デバイスからディスク `hdisk2` と `hdisk3` にボリューム・グループ・イメージを復元するには、以下のように入力します。

```
restvg -f/dev/rmt1 hdisk2 hdisk3
```

2. `/mydata/myvg` ファイルに保管されているボリューム・グループ・イメージを、バックアップ・イメージ内に入っている `vgname.data` ファイルに指定されているディスクに復元するには、以下のように入力します。

```
restvg -f/mydata/myvg
```

3. `vgname.data` ファイル `/home/my_dir/my_vg.data` だけを使用して、ファイルの復元は行わずに、ボリューム・グループの論理ボリューム構造を再作成するには、次のように入力します。

```
restvg -r -d /home/my_dir/my_vg.data
```

**注:** `vgname.data` ファイルは、`mkvgdata` コマンドを使用してボリューム・グループに作成することができます。

4. `/dev/rmt0` 内のテープにあるボリューム・グループ・バックアップの中にある、`vgname.data` ファイルを使用して、ファイルの復元を行わずに、ボリューム・グループ論理ボリューム構造を再作成するには、次のように入力します。

```
restvg -r -f /dev/rmt0
```

5. `/dev/rmt0` 内のテープにバックアップされたボリューム・グループのボリューム・グループ情報を表示するには、次のように入力します。

```
restvg -l -f /dev/rmt0
```

6. バックアップ・イメージ内に含まれる `vgname.data` ファイルで指定されたディスク上へ、`/dev/usbms0` デバイスからのボリューム・グループ・イメージを復元するには、以下のように入力します。

```
restvg -f /dev/usbms0
```



## 関連情報

`mkvgdata` コマンド、 `restore` コマンド、 `savevg` コマンド、 `mkdev` コマンド、 `reducevg` コマンド。

---

## resumevsd コマンド

### 目的

使用可能な仮想共用ディスクを活動化します。

### 構文

```
resumevsd [-p | -b | -l server_list] {-a | vsd_name ...}
```

### 説明

`resumevsd` コマンドは、指定した仮想共用ディスクを延期状態から活動状態にします。仮想共用ディスクは引き続き使用可能です。仮想共用ディスクが延期状態のために保留中であった読み取りおよび書き込み要求は再開されます。

このコマンドは、System Management Interface Tool (SMIT) を使用して実行できます。SMIT を使用するには、以下のように入力します。

```
smit vsd_mgmt
```

さらに、**Resume a Virtual Shared Disk** オプションを選択します。

通常的环境では、このコマンドを発行すべきではありません。リカバリー可能仮想共用ディスク・サブシステムはこのコマンドを使用して、制御された方法により共用ディスクを管理します。このコマンドを発行する場合、結果は予測不能です。

### フラグ

- p** グローバル・ボリューム・グループがアクティブ・サーバーとなるように定義された 1 次サーバー・ノードを指定します。**-p** フラグは CVSD に対しては有効ではありません。
- b** グローバル・ボリューム・グループに対して定義された 2 次サーバー・ノードがアクティブ・サーバーとなるように指定します。**-b** フラグは CVSD に対しては有効ではありません。
- a** 定義済みの仮想共用ディスクがすべて再開されるように指定します。
- l** **server\_list** をドライバーへ渡します。

### パラメーター

*vsd\_name* 仮想共用ディスクを指定します。

### セキュリティ

このコマンドを実行するには、**root** 権限が必要です。



## 制約事項

このコマンドはピア・ドメイン内のオンラインのノードから発行する必要があります。オンラインのピア・ドメインに移動するには **starttrpdomain** コマンドを使用します。既存のピア・ドメイン内で、オンラインの特定ノードに移動するには **starttrpnode** コマンドを使用します。RSCT ピア・ドメインの作成と管理の詳細については、「*RSCT Administration Guide*」を参照してください。

通常的环境では、このコマンドを発行すべきではありません。リカバリー可能仮想共用ディスク・サブシステムはこのコマンドを使用して、制御された方法により共用ディスクを管理します。このコマンドを発行する場合、結果は予測不能です。

## 例

仮想共用ディスク **vsd1vg1n1** を延期状態から活動状態にするには次のように入力します。

```
resumevsd vsd1vg1n1
```

## 位置

**/opt/rsct/vsd/bin/resumevsd**

## 関連情報

コマンド: **cfgvsd**、**lsvsd**、**preparevsd**、**startvsd**、**stopvsd**、**suspendvsd**、**ucfgvsd**

---

## rev コマンド

### 目的

ファイルの各行の文字の順序を逆にします。

### 構文

```
rev [ File ...]
```

### 説明

**rev** コマンドは、各行の文字の順序を逆にして、指定されたファイルを標準出力に出力します。ファイルを指定しなければ、**rev** コマンドは標準入力を読み取ります。

### 例

ファイルの各行の順序を逆にするには、以下のように入力します。

```
rev file
```

file に以下のテキストが入っている場合、

```
abcdefghi  
123456789
```

**rev** コマンドは以下のように表示します。

```
ihgfedcba  
987654321
```

## ファイル

`/usr/bin/rev` `rev` コマンドが入っています。

## 関連情報

オペレーティング・システムおよびデバイスのマネージの入出力リダイレクトのセクション。

---

## revnetgroup コマンド

### 目的

NIS マップのネットワーク・グループ・ファイル内のユーザーとホストのリスト順を逆にします。

### 構文

```
/usr/sbin/revnetgroup { -h | -u } [ File]
```

### 説明

**revnetgroup** コマンドは、ホストおよびユーザーが `/etc/netgroup` ファイル内でリストされている順序を逆にします。**revnetgroup** コマンドは、`/var/yp/Makefile` ファイルから呼び出され、NIS マップ **netgroup.byuser** または **netgroup.byhost** のいずれかを作成するための出力を生成します。この出力ファイルの各行は、ホスト名またはユーザー名とドメイン名を連結することによって形成されたキーで始まります。キーの後には、ホストまたはユーザーが属しているグループのリストが続きます。リストの先頭にはタブが付き、各グループはコンマで区切られます。

注: グループのリストには、汎用グループ (ネットワーク内のすべてのユーザーが入っているグループ) の名前は使用されません。汎用グループは \* (アスタリスク) の下にリストされます。

**revnetgroup** コマンドは、デフォルトの `/etc/netgroup` ファイルが必要でなければ、オプションのファイル名を使用します。この機能により、ユーザーはカスタム・ネットワーク・グループ・マップを作成しやすくなります。

### フラグ

**-h** **netgroup.byhost** マップを作成するための出力を生成します。  
**-u** **netgroup.byuser** マップを作成するための出力を生成します。

### 例

- `/etc/netgroup` ファイルでユーザー名がホスト名の前にリストされるようにするには、`/var/yp/Makefile` 内の該当するスタンザを以下のようにして変更します。  

```
revnetgroup -u
```
- `newgroup` という名前の新しいネットワーク・グループ・ファイルを `/etc` ディレクトリー内に作成するには、`/var/yp/Makefile` 内の該当するスタンザを以下のようにして変更します。  

```
revnetgroup -h newgroup
```

この例で使用されている **-h** フラグは、新しい `/etc/newgroup` ファイルの中でホスト名がユーザー名の前にリストされるようにします。

## ファイル

`/etc/netgroup`  
`/var/yp/Makefile`

ネットワーク・グループ内のユーザーとホストのリストが入っています。  
NIS マップの作成規則が入っています。

## 関連情報

`makedbm` コマンド、`ypinit` コマンド、`yppush` コマンド。

ネットワークおよびコミュニケーションのマネージ のネットワーク・ファイルシステム (NFS) の概要のセクション。

*AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide* の Network Information Service (NIS) のセクション。

NIS のリファレンス。

---

## rexrd デーモン

### 目的

リモート・マシン用のプログラムを実行します。

### 構文

`/usr/sbin/rpc.rexd`

### 説明

**rexrd** デーモンは、クライアントがリモート・マシン上でプログラムを実行するように要求を出すと、リモート・マシン用のプログラムを実行します。**inetd** デーモンは `/etc/inetd.conf` ファイルから **rexrd** デーモンを始動します。

非対話式プログラムは、TCP 接続に直接接続された標準ファイル・ディスクリプターを使います。対話式プログラムは、**rlogin** コマンドが提供するログイン・セッションに似た疑似端末を使用します。**rexrd** デーモンは、ネットワーク・ファイルシステム (NFS) を使ってリモート実行要求内で指定されたファイルシステムをマウントできます。診断メッセージは、一般にコンソールに出力されて要求側に戻されます。

注: root ユーザーは、**on** コマンドなどの **rexrd** クライアント・プログラムを使用してコマンドを実行することができません。

## ファイル

`/tmp_rex/rexd`  
`/etc/exports`  
`inetd.conf`  
`/etc/passwd`

リモート・ファイルシステムのための一時マウント・ポイントが入っています。  
サーバーがエクスポートできるディレクトリーをリストします。  
RPC デーモンおよびその他の TCP/IP デーモンを始動します。  
コンピューターへのログイン許可を持つユーザーごとのエントリーが入っています。

## 関連情報

**on** コマンド。

**inetd** デーモン。

ネットワークおよびコミュニケーションのマネージ のネットワーク・ファイルシステム (NFS) の概要のセクション。

ネットワークおよびコミュニケーションのマネージ の NFS コマンドのリストのセクション。

---

## rexec コマンド

### 目的

コマンドをリモート・ホスト上で一度に 1 つずつ実行します。

### 構文

**rexec** [ **-a** ] [ **-d** | **-n** ] [ **-i** ] *Host Command*

### 説明

**/usr/bin/rexec** コマンドは、指定されたリモート・ホスト上でコマンドを実行します。

**rexec** コマンドは、**\$HOME/.netrc** ファイルにリモート・ホストで使用するためのユーザー名とパスワードが入っているかどうかを検査することにより、自動ログイン機能を提供しています。このエントリーが検出されない場合や、システムがセキュア・モードで操作中の場合 (**securetcip** コマンドを参照) には、**rexec** コマンドはリモート・ホストで使用するのに有効なユーザー名とパスワードの入力を求めるプロンプトを表示します。いずれの場合も、**rexec** を使用すると、リモート・システムの **rexecd** では、ユーザーに関してデフォルトの **compat** ログイン認証メソッドが使用されます。**rexecd** は、代替認証メソッドに関して、リモート・システムの **/etc/security/user** ファイルは検査しません。**rexec** コマンド行に **-n** フラグを指定して、自動ログイン機能をオーバーライドすることもできます。

### フラグ

- a** リモート・コマンドの標準エラーを指示は、標準出力と同様です。任意のシグナルをリモート・プロセスに送信するサービスは提供されません。
- d** ソケット・レベルのデバッグを使用可能にします。
- i** **stdin** を読み取れないようにします。
- n** 自動ログインできないようにします。**-n** フラグを指定すると、**rexec** コマンドは **\$HOME/.netrc** ファイルを検索せずに、リモート・ホストで使うユーザー名とパスワードの入力を求めるプロンプトを表示します。

### パラメーター

- Command** フラグやパラメーターを含めて、リモート・ホスト上で実行すべきコマンドを指定します。
- Host** コマンドを実行すべきホストの名前を英数字フォーマットで指定します。

## 例

1. リモート・ホストで **date** コマンドを実行するには、以下のように入力します。

```
rexec host1 date
```

**date** コマンドからの出力がローカル・システムに表示されます。この例で、ローカル・ホストの **\$HOME/.netrc** ファイルには、リモート・ホスト側で有効なユーザー名とパスワードが入っています。

リモート・ホストに対して有効なエントリーが **\$HOME/.netrc** ファイルになれば、ログイン ID とパスワードの入力を求めるプロンプトが表示されます。要求されたログイン情報を入力すると、**date** コマンドの出力がローカル・システムに表示されます。

2. 自動ログイン機能をオーバーライドして、リモート・ホストで **date** コマンドを実行するには、以下のように入力します。

```
rexec -nhost1 date
```

プロンプトが表示されたらユーザー名とパスワードを入力します。

**date** コマンドからの出力がローカル・システムに表示されます。

3. リモート・ホストの別のユーザーのディレクトリーをリストするには、以下のように入力します。

```
rexec host1 ls -l /home/karen
```

リモート・ホスト **host1** のユーザー **karen** のディレクトリーのリストがローカル・システム上に表示されます。

リモート・ホストに対して有効なエントリーが **\$HOME/.netrc** ファイルになれば、ログイン ID とパスワードの入力を求めるプロンプトが表示されます。要求されたログイン情報を入力すると、リモート・ホスト **host1** 上のユーザー **karen** のディレクトリーのリストがローカル・システム上に表示されます。

## 関連情報

ネットワークおよびコミュニケーションのマネージ の通信およびネットワークのセクション。

**ftp** コマンド、**rlogin** コマンド、**rsh** コマンド、**securetcip** コマンド。

**rexecd** デーモン。

**.netrc** ファイル・フォーマット。

---

## rexecd デーモン

### 目的

**rexec** コマンド用のサーバー機能を提供します。

### 構文

注: **rexecd** デーモンは通常、**inetd** デーモンによって開始されます。また、**SRC** コマンドを使用してコマンド行から制御することもできます。

```
/usr/sbin/rexecd [ -sc ]
```

## 説明

`/usr/sbin/rexecd` デーモンは、`rexec` コマンドのサーバーです。このデーモンは、外部ホストが発行したコマンドを処理して、そのコマンドの出力を発行元の外部ホストに戻します。`rexecd` デーモンは、伝送制御プロトコル/インターネット・プロトコル (TCP/IP) 接続を通じてデータを送受信します。

`rexecd` デーモンへの変更は、Web-based System Manager、System Management Interface Tool (SMIT)、システム・リソース・コントローラー (SRC) を使用し、`/etc/inetd.conf` または `/etc/services` ファイルを編集することで行うことができます。コマンド行で `rexecd` と入力することはお勧めできません。`/etc/inetd.conf` ファイル内にコメントが付けられていなければ、`rexecd` デーモンはデフォルトで始動します。

`inetd` デーモンは、`/etc/inetd.conf` ファイルと `/etc/services` ファイルの情報を取得します。

`etc/inetd.conf` ファイルを変更したら、`refresh -s inetd` または `kill -1 InetdPID` コマンドを実行して、構成ファイルを変更したことを `inetd` デーモンに通知します。

注: `rexecd` デーモンは無効なオプションを無視します。`syslog` 機能が使用可能になっている場合は、情報はシステム・ログに記録されます。

## フラグ

- c 逆ネーム・レゾリューションを行えないようにします。`-c` フラグを指定しなかった場合に、クライアントの逆ネーム・レゾリューションが失敗すると、`rexecd` デーモンは失敗します。
- s ソケット・レベルのデバッグを使用可能にします。

## サービス要求プロトコル

`rexecd` デーモンは、要求を受信すると以下のプロトコルを開始します。

1. サーバーはソケットから null (0) バイトまでの文字を読み取り、読み取った文字列を ASCII 数値 (10 進) として解釈します。
2. 受信した数値がゼロ以外であれば、`rexecd` デーモンは、標準エラー出力に使う 2 次ストリームのポート番号として解釈します。その後、`rexecd` デーモンはクライアント・マシンに 2 番目の指定のポート接続を作成します。
3. `rexecd` デーモンは、初期ソケット上で最高 16 文字までの末尾が null のユーザー名を検索します。

## セキュリティ

`rexecd` デーモンは、`rexec` というサービス名を持つ、PAM 使用可能アプリケーションです。認証に PAM を使用するシステム規模の構成は、`/etc/security/login.cfg` の `usw` スタンザにある `auth_type` 属性の値を、root ユーザーと同じ `PAM_AUTH` に変更することにより、設定されます。

PAM が使用可能であるときに使用される認証メカニズムは、`/etc/pam.conf` 内の `rexec` サービスの構成によって決まります。`rexecd` デーモンは、`auth` モジュール・タイプ、`account` モジュール・タイプ、および `session` モジュール・タイプの `/etc/pam.conf` エントリーを必要とします。以下にリストされるのは、`rexec` サービスにおける `/etc/pam.conf` 内の推奨される構成です。

```
#
# AIX rexec configuration
#
rexec auth      required      /usr/lib/security/pam_aix
```

```
rexec account    required    /usr/lib/security/pam_aix
rexec session    required    /usr/lib/security/pam_aix
```

## 関連情報

**kill** コマンド、 **lssrc** コマンド、 **refresh** コマンド、 **rexec** コマンド、 **startsrc** コマンド、 **stopsrc** コマンド。

**inetd** デーモン。

**rexec** サブルーチン。

**/etc/inetd.conf** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ の TCP/IP のセクション。

Web-based System Manager のインストールについては、 *AIX 5L Version 5.3 Web-based System Manager 管理ガイド* の第 2 章 インストールおよびシステム要件のセクションを参照してください。

ネットワークおよびコミュニケーションのマネージ の TCP/IP デーモンのセクション。

---

## rgb コマンド

### 目的

X-Window システム・サーバーが使う色のデータベースを作成します。

### 構文

```
rgb [ DatabaseName ] [ <InputfileName > ]
```

### 説明

**rgb** コマンドは、標準入力から行を読み取り、データベースに挿入して、カラー名と特定の赤、緑、青 (RGB) の値を関連付けます。

**rgb** コマンドは、 *DatabaseName.dir* と *DatabaseName.pag* という 2 つの出力ファイルを生成します。データベース・ファイル名を指定しなければ、デフォルト名 **rgb.dir** と **rgb.pag** が使われます。

各カラー・エントリーのフォーマットは以下のとおりです。

```
Red Green Blue Colorname
```

*Red*、*Green*、*Blue* の各エレメントは、0 から 255 までの整数値です。実際の色は、エレメントの組み合わせ方によって決まります。各エレメントは、輝度なし (0) から全輝度 (255) の範囲で指定できます。

*Colorname* パラメーターには、記述的または空想的な名前を指定できます。例えば、シーケンス 250 250 250 に、white または snow という名前を与えることができます。複数のエントリーに、同じエレメント番号またはエレメント名を共用させることもできます。

### パラメーター

|                          |                          |
|--------------------------|--------------------------|
| <i>DatabaseName</i>      | 出力データ用に作成するデータベースを指定します。 |
| < <i>InputFileName</i> > | 入力ファイル名を指定します。           |



## 例

1. 下記の例は、入力ファイルの一部を示しています。

```
248 248 255      ghost white
245 245 245      white smoke
255 250 240      floral white
253 245 230      old lace
250 240 230      linen
255 218 185      peach puff
255 248 220      cornsilk
255 250 205      lemon chiffon
245 255 250      mint cream
240 255 255      azure
```

2. 下記の例では、出力ファイル **Newcolor.dir** および **Newcolor.pag** が生成されます。

```
rgb Newcolor < rgb.txt
```

**Newcolor** は *DatabaseName* であり、**rgb.txt** は *InputFileName* です。

## ファイル

`/usr/lib/X11/rgb.txt`

デフォルトの rgb データベース入力ファイル。

---

## ripquery コマンド

### 目的

RIP ゲートウェイを照会します。

### 構文

```
ripquery [ -1 ] [ -2 ] [ -[a5] authkey ] [ -n ] [ -N dest[/mask] [ -p ] [ -r ] [ -v ] [ -w time ] gateway...
```

### 説明

**ripquery** コマンドは、RIP ゲートウェイ に認識されているすべての経路指定を **RIP REQUEST** または **POLL** コマンドを送信して要求する場合に使用されます。戻された経路指定パケットに入っているルーティング情報は、数値と記号で表示されます。 **ripquery** コマンドは、ネットワーク管理ではなく、ゲートウェイ をデバッグするためのツールとして使用されるのを目的とします。SNMP は、ネットワーク管理の優先プロトコルです。

### フラグ

|                      |                                                                                                                                                                                       |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-1</b>            | 照会をバージョン 1 パケットとして送信します。                                                                                                                                                              |
| <b>-2</b>            | 照会をバージョン 2 パケットとして送信します (デフォルト)。                                                                                                                                                      |
| <b>-[a5] authkey</b> | 照会に使用する認証パスワードを指定します。 <b>-a</b> を指定すると認証タイプ <b>SIMPLE</b> が使用され、 <b>-5</b> を指定すると認証タイプ <b>MD5</b> が使用されます。その他の場合は、デフォルトの認証タイプ <b>NONE</b> が使用されます。着信パケットの認証フィールドが表示されますが、妥当性は検査されません。 |
| <b>-n</b>            | シンボル名を判別するために、応答ホストのアドレスが調べられないようにします。                                                                                                                                                |

|                              |                                                                                                                                                                                                                                                                       |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-N</b> <i>dest[/mask]</i> | これを指定すると、照会が、完全な経路指定テーブルではなく、指定された <i>dest/mask</i> に対して行われます。オプションのマスクの指定は、バージョン 2 の照会を暗黙指定します。特定の出力先に関して、最大 23 までの要求を 1 つのパケットの中に組み込むことができます。                                                                                                                       |
| <b>-p</b>                    | 経路指定テーブルの情報を要求するのに <b>RIP POLL</b> コマンドを使用します。これはデフォルトです。 <b>RIP POLL</b> コマンドへの応答がないと、 <b>RIP REQUEST</b> コマンドが試行されます。 <b>gated</b> は、RIP を介して得たすべての経路を表示して <b>POLL</b> コマンドに応答します。                                                                                  |
| <b>-r</b>                    | ゲートウェイの経路指定テーブルの情報を要求するのに <b>REQUEST</b> コマンドを使用します。 <b>RIP POLL</b> コマンドとは異なり、すべてのゲートウェイで <b>RIP REQUEST</b> をサポートする必要があります。 <b>RIP REQUEST</b> コマンドへの応答がないと、 <b>RIP POLL</b> コマンドが試行されます。 <b>gated</b> は、指定したインターフェースを介して告知したすべての経路を表示して <b>REQUEST</b> コマンドに応答します。 |
| <b>-v</b>                    | <b>ripquery</b> に関するバージョン情報が表示されてから、ゲートウェイが照会されます。                                                                                                                                                                                                                    |
| <b>-w</b> <i>time</i>        | ゲートウェイからの最初の応答の待ち時間を秒単位で指定します。デフォルト値は 5 秒です。                                                                                                                                                                                                                          |

## 関連情報

**gated** デーモン。

---

## rksh コマンド

### 目的

Korn シェルの制限付きバージョンを起動します。

### 構文

```
rksh [ -i ] [ { + | - } { a e f h k m n p t u v x } ] [ -o Option ... ] [ -c String | -s | File [ Parameter ] ]
```

注: フラグの前に - (マイナス) ではなく + (プラス) を付けると、そのフラグをオフにします。

### 説明

**rksh** コマンドは、Korn シェルの制限付きバージョンを起動します。これにより管理者は、ユーザーに対して制御されたシェル環境を提供することができます。拡張 Korn シェルの場合は、**rksh93** と呼ばれる、**rksh** の制限付きバージョンも使用可能です。

制限付きシェルでは、ユーザーは次のことは行えません。

- 現在の作業ディレクトリーの変更。
- SHELL、ENV または PATH 変数の設定。
- / (スラッシュ) を含むコマンドのパス名の指定。
- > (右脱字記号)、>| (右脱字記号、パイプ記号)、<> (左脱字記号、右脱字記号)、または >> (2 つの右脱字記号) を使用するコマンドの出力の、リダイレクト。

制限付き Korn シェルについて詳しくは、オペレーティング・システムおよびデバイスのマネージの制限付き Korn シェルを参照してください。

## フラグ

- a** 定義されているすべての後続パラメーターを自動的にエクスポートします。
- c *String*** Korn シェルに、*String* 変数からコマンドを読み取らせませす。このフラグは、**-s** フラグあるいは *File[Parameter]* パラメーターとともに使用することはできません。
- e** **ERR** トラップが設定されていれば実行し、コマンドがゼロ以外の終了状況であれば終了します。このモードは、プロファイルの読み取り中は使用不可です。
- f** ファイル名の置換を使用不可にする。
- h** 各コマンドが最初に行われたとき、トラッキングされる別名を指定します。
- i** シェルが対話式であることを示します。対話式シェルは、シェル入出力が端末に接続されるかどうかを示します (**ioctl** サブルーチンによって決まるとおりです)。この場合、**TERM** 環境変数は無視され (このため **kill 0** コマンドは対話式シェルを **kill** しません)、**INTR** シグナルが受け取られて無視されます (このため待ち状態が中断される場合があります)。すべての場合において、シェルは **QUIT** シグナルを無視します。
- k** コマンド名の前にあるパラメーター割り当て引数のみでなく、環境内のすべてのパラメーター割り当て引数をコマンドに配置します。
- m** 別のプロセス内でバックグラウンド・ジョブを実行し、完了時に 1 行を印刷します。バックグラウンド・ジョブの終了状況が、完了メッセージの中で報告されます。ジョブ制御付きのシステムでは、このフラグは対話式シェルの場合は自動的にオンになります。
- n** コマンドを読み取り、構文エラーがないか検査しますが、コマンドの実行はしません。このフラグは、対話式シェルの場合は無視されます。

## -o Option

現行のオプション設定、およびユーザーが引数を指定しない場合はエラー・メッセージを、印刷します。このフラグを使用して、次のオプションのいずれかを使用可能にすることができます。

### **allexport**

**-a** フラグと同じです。

**errexit -e** フラグと同じです。

**bgnice** 低い優先順位でバックグラウンド・ジョブを実行します。これがデフォルトのモードです。

**emacs** コマンド・エンタリーに、**emacs-** スタイルのインライン・エディターを入力します。

**gmacs** コマンド・エンタリーに **gmacs-** スタイルのインライン・エディターを入力します。

### **ignoreeof**

ファイル・マークが現れたときにシェルを終了しません。 **exit** コマンドを使用するか、または **Ctrl-D** キー・シーケンスを 11 回より多く押すことによりフラグをオーバーライドしてシェルを終了する必要があります。

### **keyword**

**-k** フラグと同じです。

### **markdirs**

ファイル名置換の結果であるすべてのディレクトリー名に、 **/** (スラッシュ) を付加します。

### **monitor**

**-m** フラグと同じです。

### **noclobber**

リダイレクトにより既存ファイルが切り捨てられないようにします。このオプションを指定する場合、ファイルを切り捨てるにはリダイレクト・シンボル **>|** (右脱字記号、パイプ・シンボル) を使用します。

### **noexec**

**-n** フラグと同じです。

**noglob -f** フラグと同じです。

**nolog** ファンクション定義がヒストリー・ファイルに保管されないようにします。

### **nounset**

**-u** フラグと同じです。

### **privileged**

**-p** フラグと同じです。

### **verbose**

**-v** フラグと同じです。

### **trackall**

**-h** フラグと同じです。

**vi** コマンド・エンタリーに、**vi-style** インライン・エディターの挿入モードを入力します。エスケープ文字 **033** を入力することにより、エディターは移動モードに置かれます。戻りによりこの行が送信されます。

**viraw** 各キャラクターを、**vi** モードで入力されたとおりに処理します。

**xtrace -x** フラグと同じです。

単一の **rksh** コマンド行に複数のオプションを設定することができます。

|           |                                                                                                                                                             |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-p</b> | ログイン・シェルとしてシェルを使用時に、 <b>\$HOME/.profile</b> ファイルの処理を使用不可にします。                                                                                               |
| <b>-s</b> | <b>rksh</b> コマンドは標準入力からコマンドを読み取ります。特別なコマンドの出力を除き、シェル出力はファイル・ディスクリプター 2 に書き込まれます。このパラメーターは、 <b>-c</b> フラグあるいは <i>File[Parameter]</i> パラメーターとともに使用することはできません。 |
| <b>-t</b> | 1 つのコマンドを読み込み、実行後、終了します。                                                                                                                                    |
| <b>-u</b> | 置換の際に、設定解除パラメーターをエラーとして扱います。                                                                                                                                |
| <b>-v</b> | シェル入力行を、読み込まれたとおりに印刷します。                                                                                                                                    |
| <b>-x</b> | 実行されたコマンドおよびその引数を印刷します。                                                                                                                                     |

## ファイル

**/usr/bin/rksh**                   制限付き Korn シェルへのパス名が入っています。  
**/tmp/sh\***                       シェルがオープンした時に作成された一時ファイルを含みます。

## 関連情報

**env** コマンド。

**ksh** コマンドおよび **rksh93** コマンド。

ネットワークおよびコミュニケーションのマネージ の通信およびネットワークのセクション。

オペレーティング・システムおよびデバイスのマネージの制限付き Korn シェルおよびシェルのセクション。

---

## rlogin コマンド

### 目的

ローカル・ホストをリモート・ホストと接続します。

### 構文

**rlogin RemoteHost** [ **-e Character** ] [ **-8** ] [ **-l User** ] [ **-f | -F** ] [ **-k realm** ]

### 説明

**/usr/bin/rlogin** コマンドは、指定されたりモート・ホストにログインして、ローカル端末をリモート・ホストに接続します。

リモート端末のタイプは、**TERM** ローカル環境変数で指定されたタイプと同じです。端末またはウィンドウ・サイズは、リモート・ホストがサポートする場合は同じになり、変更はすべて転送されます。すべてのエコーはリモート・ホスト側で発生するので、遅延を除き端末接続は透過です。 **Ctrl-S** および **Ctrl-Q** キー・シーケンスは情報の流れを停止および開始させ、入出力バッファは割り込み時にフラッシュされます。

### リモート・コマンドの実行

**rlogin** コマンドを使うと、ホスト名をリンク名として使用してパスへのリンクを作成できます。以下に例を示します。

```
ln -s /usr/bin/rsh HostName
```

プロンプトが表示されたときに引数 (コマンド) が付く *HostName* パラメーターで指定したホスト名を入力すると、自動的に **rsh** コマンドを使用して、*HostName* パラメーターで指定したリモート・ホストのコマンド行に指定されるコマンドをリモートで実行することになります。

プロンプトが表示されたときに引数 (コマンド) の付かない *HostName* パラメーターで指定したホスト名を入力すると、自動的に **rlogin** コマンドを使用して、*HostName* パラメーターで指定したリモート・ホストにログインすることになります。

上記の条件の他に、リモート・ユーザーのアカウントにパスワードが定義されていなければ、**rlogin** コマンドを使用してリモート・ホストにアクセスできます。ただし、セキュリティ上の理由で、すべてのユーザー・アカウント上でパスワードを使用することをお勧めします。

**rlogin** コマンドは、(**exec** コマンドを使用して) **/usr/sbin/login** ファイルを実行し、ユーザーの妥当性を検査します。これにより、1) すべてのユーザー属性およびデバイス属性が **telnet** 接続に影響を与え、2) 一度に可能なログイン・セッションの最大数 (**maxlogins** 属性で決まる) に対してリモート・ログイン数をカウントすることができます。属性の定義は **/etc/security/user** ファイルおよび **/etc/security/login.cfg** ファイルで行います。

## POSIX 伝送制御手順

**rlogind** デーモンと **telnetd** デーモンは、POSIX 伝送制御手順を使用してローカル TTY 上の伝送制御手順を変更します。ローカル TTY 上で POSIX 伝送制御手順が使われていない場合に、他の伝送制御手順をエコーすると、正しく動作しないことがあります。TCP/IP が正常に機能するためには、POSIX 伝送制御手順がなければなりません。

## フラグ

- 8** 常に 8 ビット・データ・パスが使用できるようにします。そうでない場合、**rlogin** コマンドは 7-bit データ・パスを使用し、パリティ・ビットは取り除かれます。ただし、リモート・ホスト上の開始文字と停止文字が **Ctrl-S** と **Ctrl-Q** でない場合を除きます。
- e Character** エスケープ文字を変更します。*Character* に選択した文字を入れてください。
- f** これを指定すると、証明書が転送されます。このフラグは、**Kerberos 5** が現在の認証メソッドでない場合は無視されます。認証は、現在の **DCE** 証明書に転送可能のマークが付けられていない場合には無視されます。
- F** これを指定すると、証明書が転送されます。さらに、リモート・システムの証明書に転送可能のマークが付けられます (これによって、別のリモート・システムにそれらの証明書を渡すことができるようになります)。このフラグは、**Kerberos 5** が現在の認証メソッドでない場合は無視されます。認証は、現在の **DCE** 証明書に転送可能のマークが付けられていない場合には無視されます。
- k realm** これを使用することで、ユーザーは、リモート端末のレルムを指定できます。ただし、そのレルムが **DCE** セルと同意の場合に限ります。このフラグは、**Kerberos 5** が現在の認証メソッドでない場合は無視されます。
- l User** リモート・ユーザー名を指定する名前に変更します。または、ローカル・ユーザー名がリモート・ホスト側で使用されます。

## セキュリティ

複数の認証メソッドがありますが、それぞれのメソッドでは、接続を可能にするために別々の事項を設定する必要があります。

## 標準の認証の場合

リモート・ホストは、以下の条件の一方または両方が満たされる場合にのみアクセスできます。

- ローカル・ホストがリモートの `/etc/hosts.equiv` ファイル内に含まれ、ローカル・ユーザーが `root` ユーザーでなく、`-iUser` フラグが指定されていない場合。
- ローカル・ホストおよびユーザー名が、リモート・ユーザー・アカウントにある `$HOME/.rhosts` ファイルに含まれている場合。

ユーザーは `$HOME/.rhosts` ファイルに対して、許可を任意に設定できますが、`.rhosts` ファイルの許可は 600 (オーナーだけが読み取りおよび書き込み可能) に設定することをお勧めします。

**注:** `AUTHSTATE` 環境変数は、ユーザーが認証されるレジストリーを示します。例えば、LDAP サーバー上で定義された LDAP ユーザーは、パスワードを使用してリモート・システムにログインした場合、`AUTHSTATE` が LDAP に設定されます。しかし、ユーザーが `$HOME/.rhosts` ファイルおよび `/etc/hosts.equiv` ファイルのエントリーを通して認証される場合は、そのユーザーの `AUTHSTATE` 環境変数は、ユーザー ID が定義されているかどうかに関係なく、`compat` に設定されます。

## Kerberos 5 認証の場合

次の条件のすべてが満たされた場合にのみ、リモート・ホストへのアクセスが可能になります。

- ローカル・ユーザーに現在の DCE 証明書が与えられている。
- ローカル・システムとリモート・システムが、Kerberos 5 認証に対して構成されている (一部のリモート・システムの場合、これは不要のことがある。デーモンが `klogin` ポートを `listen` していることが必要)。
- リモート・システムが DCE 証明書をリモート・アカウントへのアクセスに充分であるとして受け入れる。追加情報については、`kvalid_user` 関数を参照のこと。

**-e** フラグによる変更がない限り、標準エスケープ文字はティルド (`~`) です。エスケープ文字が行頭にあると、リモート・ホストにしか認識されません。行頭以外の位置にあるエスケープ文字は、通常文字としてリモート・ホストに送信されます。エスケープ文字を行頭にある通常文字としてリモート・ホストに送信するには、エスケープ文字を 2 つ入力します。エスケープ文字とピリオド (例えば `~.`) を押すと、すぐにリモート・ホストからローカル端末を切り離します。

## 例

- ローカル・ユーザー名を使用してリモート・ホストにログインするには、以下のように入力します。

```
rlogin host2
```

パスワードの入力を求めるプロンプトが表示されてから、リモート・ホスト `host2` にログインします。

リモート・ホストをログオフして接続をクローズするには、`~.` (ティルド、ピリオド) を入力します。

- 別のユーザー名を使用してリモート・ホストにログインするには、以下のように入力します。

```
rlogin host2 -l dale
```

パスワードの入力を求めるプロンプトが表示されてから、ユーザー名 `dale` が付いたリモート・ホスト `host2` にログインします。

リモート・ホストをログオフして接続をクローズするには、`~.` (ティルド、ピリオド) を入力します。

- ローカル・ユーザー名を使用してリモート・ホストにログインしエスケープ文字を変更するには、以下のように入力します。



```
rlogin host2 -e\
```

パスワードの入力を求めるプロンプトが表示されてから、リモート・ホスト `host2` にログインします。エスケープ文字は `\` (バックスラッシュ) に変更されています。

リモート・ホストをログオフにして接続をクローズするには、`\.` (バックスラッシュ、ピリオド) を入力します。

## 関連情報

**ftp** コマンド、**login** コマンド、**rcp** コマンド、**rexec** コマンド、**rsh** コマンド、**telnet** コマンド。

**rlogind** および **krlogind** デーモン。

**kvalid\_user** 関数。

**hosts.equiv** ファイル・フォーマット、**.rhosts** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージの通信およびネットワークのセクション。

ネットワークおよびコミュニケーションのマネージの認証およびセキュア **rcmd** のセクション。

---

## rlogind デーモン

### 目的

**rlogin** コマンドのサーバー機能を提供します。

### 構文

注: **rlogind** デーモンは通常、**inetd** デーモンにより開始されます。また、**SRC** コマンドを使用してコマンド行から制御することもできます。

```
/usr/sbin/rlogind [ -a ] [ -c ] [ -l ] [ -n ] [ -s ]
```

### 説明

**/usr/sbin/rlogind** デーモンは、**rlogin** リモート・ログイン・コマンドのサーバーです。このサーバーはリモート・ログイン機能を提供します。

**rlogind** デーモンへの変更は、Web-based System Manager、System Management Interface Tool (SMIT)、システム・リソース・コントローラー (SRC) を使用し、**/etc/inetd.conf** または **/etc/services** ファイルを編集することで行うことができます。コマンド行に **rlogind** と入力することはお勧めできません。

**rlogind** デーモンは、**/etc/inetd.conf** ファイル内でコメントされていないときには、デフォルトで始動されます。

**rlogind** デーモンは認識されないオプションを無視します。システムで **syslog** サービスが使用可能になっている場合は、**syslog** サービスを使用してこの情報を記録します。

**inetd** デーモンは、**/etc/inetd.conf** ファイルと **/etc/services** ファイルから情報を取り出します。

**/etc/inetd.conf** ファイルまたは **/etc/services** ファイルを変更したら、**refresh -s inetd** または **kill -1 InetdPID** コマンドを実行して、構成ファイルを変更したことを **inetd** デーモンに通知します。

## サービス要求プロトコル

**rlogind** デーモンは、サービス要求を受信すると、以下のプロトコルを開始します。

1. **rlogind** デーモンは、要求についてソース・ポート番号を検査します。ポート番号が 512 から 1023 の範囲になれば、**rlogind** デーモンは接続を終了します。
2. **rlogind** デーモンは、初期接続要求のソース・アドレスを使ってクライアント・ホスト名を判別します。名前を判別できなければ、**rlogind** デーモンはクライアント・ホスト・アドレスの小数点付き 10 進数を使用します。

## エラー・メッセージ

**rlogind** デーモンに関連するエラー・メッセージは以下のとおりです。

**Try again.**

**/usr/bin/shell:**

サーバーが **fork** コマンドを実行しましたが失敗しました。

シェルがありません。シェル変数に指定されたシェルは始動できません。シェル変数にはプログラムも使えます。

## フラグ

- a **pty** 印刷速度増加機構を使用不可にします。
- c ホスト名ルックアップの健全性チェックを抑制します。
- l ユーザーの **\$HOME/.rhosts** ファイルに基づく認証を防止します。ただし、**/etc/passwd** ファイルで指定した **.rhosts** ファイルが root のホーム・ディレクトリーにあるときには、root ユーザーが自動的にログインされます。
- n 転送レベルの keep-alive メッセージを使用不可にします。デフォルトではメッセージは使用可能です。
- s ソケット・レベルのデバッグをオンにします。

## セキュリティ

**rlogind** デーモンは、**rlogin** というサービス名を持つ、PAM 使用可能アプリケーションです。認証に PAM を使用するシステム規模の構成は、**/etc/security/login.cfg** の **usw** スタanzasにある **auth\_type** 属性の値を、root ユーザーと同じ **PAM\_AUTH** に変更することにより、設定されます。

PAM が使用可能であるときに使用される認証メカニズムは、**/etc/pam.conf** 内の **rlogin** サービスの構成によって決まります。**rlogind** デーモンは、**auth** モジュール・タイプ、**account** モジュール・タイプ、**password** モジュール・タイプ、および **session** モジュール・タイプの **/etc/pam.conf** エントリーを必要とします。以下にリストされるのは、**rlogin** サービスにおける **/etc/pam.conf** 内の推奨される構成です。

```
#
# AIX rlogin configuration
#
rlogin auth      sufficient /usr/lib/security/pam_rhosts_auth
rlogin auth      required   /usr/lib/security/pam_aix

rlogin account  required   /usr/lib/security/pam_aix

rlogin password required   /usr/lib/security/pam_aix

rlogin session  required   /usr/lib/security/pam_aix
```

## 例

注: **rlogind** デーモンの引数は、SMIT を使用するか、または **/etc/inetd.conf** ファイルを編集することによって指定できます。

1. **rlogind** デーモンを始動するには、以下のように入力します。

```
startsrc -t rlogin
```

このコマンドは **rlogind** サブサーバーを始動します。

2. **rlogind** デーモンを通常通り終了するには、以下のように入力します。

```
stopsrc -t rlogin
```

このコマンドにより、保留状態の接続がすべて開始され、既存の接続が完了します。ただし、新しい接続の開始は行われません。

3. **rlogind** デーモンおよびすべての **rlogind** 接続を強制的に停止するには、以下のように入力します。

```
stopsrc -t -f rlogin
```

このコマンドにより、保留状態の接続と既存の接続がすべて即座に終了します。

4. **rlogind** デーモンに関する簡潔な状況レポートを表示するには、以下のように入力します。

```
lssrc -t rlogin
```

このコマンドは、デーモンの名前、プロセス ID、および状態 (アクティブか非アクティブか) を戻します。

## 関連情報

**kill** コマンド、**lssrc** コマンド、**refresh** コマンド、**rlogin** コマンド、**startsrc** コマンド、**stopsrc** コマンド。

**rshd** デーモン、**inetd** デーモン、**syslogd** デーモン。

**pty** スペシャル・ファイル。

**\$HOME/.rhosts** ファイル・フォーマット、**/etc/hosts.equiv** ファイル・フォーマット、**/etc/inetd.conf** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ の TCP/IP デーモンのセクション。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

---

## rm コマンド

### 目的

ファイルまたはディレクトリーを除去 (リンク解除) します。

### 構文

```
rm [ -f ] [ -r ] [ -R ] [ -i ] [ -e ] File ...
```

## 説明

**rm** コマンドは、指定した *File* パラメーターのエントリーをディレクトリーから除去します。エントリーがファイルへの最終リンクであれば、このファイルが削除されます。あるファイルに対する書き込み許可がなく、かつ標準入力端末である場合は、そのファイル名が示され、そのファイルを削除したいかどうかの確認を要求されます。 *y* (*yes* を意味する) を入力するとそのファイルが削除され、その他の任意の文字を入力するとそのファイルは削除されません。削除したいファイルの読み取り許可または書き込み許可は必要ありません。ただし、そのファイルが入っているディレクトリーの書き込み許可がなければなりません。

ファイルがシンボリック・リンクであれば、そのリンクは除去されますが、そのシンボリック・リンクが参照するファイルまたはディレクトリーはそのままの状態です。ディレクトリーへの書き込み許可があれば、シンボリック・リンクを削除するための書き込み許可は必要ありません。

.(ドット) または .. (ドット、ドット) ファイルが *File* パラメーターのベース名部分として指定されると、**rm** コマンドは標準エラーに診断メッセージを書き込み、このようなパラメーターが指定されても何も実行しないようにします。

**-f** フラグを指定していないときに、*File* パラメーターに書き込み許可がなく標準入力端末がワークステーションである場合か、または **-i** フラグを指定している場合には、**rm** コマンドは、標準エラーにプロンプトを書き込み、標準入力から 1 行読み取ります。肯定応答を入力しなければ、**rm** コマンドは現行ファイルにはそれ以上何も実行せず、次のファイルに進みます。

ディレクトリーにスティッキー・ビットが設定されていて、そのディレクトリーがユーザー所有のものでない場合には、他のユーザーの所有するファイルを除去することはできません。

**注:** **rm** コマンドは、*-* (ダッシュ、ダッシュ) パラメーターをフラグの終わりを示す区切り文字としてサポートします。

NFS バージョン 4 サーバー用にエクスポートされたファイルまたはディレクトリーを除去しようとする時、失敗して、そのリソースが使用中である旨のメッセージが出されます。ファイルまたはディレクトリーは、NFS バージョン 4 用にアンエクスポートしてから、除去する必要があります。

## フラグ

- e** 各ファイルの削除後にメッセージを表示します。
- f** 書き込み保護ファイルを除去する前にプロンプトを表示しません。指定されたファイルが存在しなくても、エラー・メッセージを表示せず、エラー状況を戻しません。 **-f** と **-i** フラグの両方が指定されている場合には、最後に指定した方に影響があります。
- i** 各ファイルを削除する前にプロンプトを表示します。 **-i** と **-r** フラグを一緒に使うと、**rm** コマンドはディレクトリーを削除する前にもプロンプトを表示します。 **-i** と **-f** フラグの両方が指定されている場合には、最後に指定した方に影響があります。
- r** *File* パラメーターにディレクトリーを指定すると、ディレクトリーとそれらの内容を再帰的に除去できます。このフラグは **-R** フラグと同等です。
- R** *File* パラメーターにディレクトリーを指定すると、ディレクトリーとそれらの内容を再帰的に除去できます。このフラグは **-r** フラグと同等です。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0 **-f** フラグを指定しなかった場合は、名前の付いたディレクトリーのエンタリーすべてが除去されるか、既存の名前の付いたディレクトリーのエンタリーがすべて除去されます。
- >0 エラーが発生しました。

## 例

1. ファイルを削除するには、以下のように入力します。

```
rm myfile
```

このファイルに別のリンクがある場合には、ファイルはそのリンクの名前で残りますが、`myfile` という名前は除去されます。`myfile` が唯一のリンクであれば、ファイル自体が削除されます。

2. 最初に確認プロンプトを受け取らずにファイルを削除するには、以下のように入力します。

```
rm -f core
```

**rm -f** コマンドが `core` という名前のファイルを除去しようとする前には、確認プロンプトは発行されません。ただし、`core` ファイルが書き込み保護されていて、ユーザーがこのファイルのオーナーでもなく、`root` 権限を持っていない場合は、エラー・メッセージが表示されます。**rm -f** コマンドが存在していないファイルを除去しようとしたときは、エラー・メッセージは表示されません。

3. ファイルを 1 つずつ削除するには、以下のように入力します。

```
rm -i mydir/*
```

それぞれのファイル名が表示されたら、削除したいファイルには `y` を入力し、保持しておきたいファイルには `Enter` キーを押します。

4. ディレクトリー・ツリーを削除するには、以下のように入力します。

```
rm -ir manual
```

このコマンドは、`manual` ディレクトリーのすべてのサブディレクトリーの内容を再帰的に除去し、それぞれのファイルの除去についてのプロンプトを表示してから、`manual` ディレクトリー自体を除去します。その例を以下に示します。

```
You: rm -ir manual
System: rm: Select files in directory manual? Enter y for yes.
You: y
System: rm: Select files in directory manual/draft1? Enter y for yes.
You: y
System: rm: Remove manual/draft1?
You: y
System: rm: Remove manual/draft1/chapter1?
You: y
System: rm: Remove manual/draft1/chapter2?
You: y
System: rm: Select files in directory manual/draft2? Enter y for yes.
You: y
System: rm: Remove manual/draft2?
You: y
System: rm: Remove manual?
You: y
```

この例では、**rm** コマンドは最初に、`manual` ディレクトリーの検索が必要かどうか尋ねます。`manual` ディレクトリーには複数のディレクトリーが入っているため、**rm** コマンドは次に、削除したいファイ

ルを `manual/draft1` ディレクトリー内で検索する許可を求めてから、`manual/draft1/chapter1` ファイルと `manual/draft1/chapter2` ファイルを削除したいかどうか尋ねます。次に `rm` コマンドは `manual/draft2` ディレクトリー内の検索の許可を求めます。そして、`manual/draft1`、`manual/draft2`、`manual` の各ディレクトリーを削除する許可を求めます。

サブディレクトリー (例えば、`manual/draft2`) を除去する許可を否認すると、`rm` コマンドは、`manual` ディレクトリーを除去しません。その代わりに、メッセージ「`rm: Directory manual not empty`」が表示されます。

## ファイル

`/usr/bin/rm` `rm` コマンドが入っています。

## 関連情報

`ln` コマンド。

`unlink` サブルーチン。

オペレーティング・システムおよびデバイスのマネージのディレクトリーのセクション。

オペレーティング・システムおよびデバイスのマネージのファイルとディレクトリーのアクセス・モードのセクション。

オペレーティング・システムおよびデバイスのマネージのファイルとディレクトリーのリンクのセクション。

---

## rm\_nisccachemgr コマンド

### 目的

`nis_cachemgr` デーモンを停止し、`/etc/rc.nfs` ファイルのエントリーにコメントを付けます。

### 構文

`rm_nisccachemgr` [ `-I` | `-B` | `-N` ]

### 説明

`rm_nisccachemgr` コマンドは、`/etc/rc.nfs` ファイルの `nis_cachemgr` デーモンのエントリーにコメントを付けます。`rm_nisccachemgr` デーモンは、`stopsrc` コマンドを使って `nis_cachemgr` デーモンを停止します。

注: `mk_nisd`、`mk_cachemgr`、`mk_nispasswd`、`rm_nisd`、`rm_cachemgr`、および `rm_nispasswd` コマンドは、次の 2 つのことを行います。

- `/etc/rc.nfs` のデーモン始動コールのエントリーを変更する。
- デーモン `src` エンティティーのデフォルトの動作を変更する。

例えば、`rpc.nisd` デーモンが `-Y` フラグによって始動されるとすると、このことは、`rpc.nisd` デーモンを始動するために `/etc/rc.nfs` エントリーでは明示的に設定されません。その代わりに、`chssys`

が実行されて、始動時にデーモンに追加されるデフォルト・オプションがあればそのオプションが入れます。これらのオプションが存在するか確認するには、**lssrc -S -s subsystem** コマンドを使用して、デフォルト・オプションを表示させます。

## フラグ

- I** **/etc/rc.nfs** ファイルの **nis\_cachemgr** デーモンのエントリーにコメントを付けます。
- B** **/etc/rc.nfs** ファイルの **nis\_cachemgr** デーモンのエントリーにコメントを付け、**nis\_cachemgr** デーモンを停止します。このフラグがデフォルトです。
- N** **stopsrc** コマンドを使用して **nis\_cachemgr** デーモンを停止します。このフラグは **/etc/rc.nfs** ファイルを変更しません。

**注:** このコマンドを実行した場合の結果として重要なことは、NIS+ 環境のドメイン名が NULL にリセットされることです。管理担当者が **nis\_cachemgr** をオフにした場合、NIS+ 構成がアクティブでなくなります。ドメイン名をリセットすると、不必要なルックアップが避けられます。管理担当者がこの結果を望まない場合、**rm\_niscachemgr** が実行された後に **chypdom** を実行しなくてはなりません。

## 例

**nis\_cachemgr** デーモンを始動する **/etc/rc.nfs** ファイル内のエントリーにコメントを付けるには、次のように入力します。

```
rm_niscachemgr -I
```

このコマンドは、現在実行中のデーモンは停止しません。

## ファイル

**/etc/rc.nfs** NFS と NIS デーモンの始動スクリプトが入っています。

## 関連情報

**smit** コマンド。

**nis\_cachemgr** デーモン。

*AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide* の Network Information Service (NIS+) のセクション。

ネットワークおよびコミュニケーションのマネージ のネットワーク・ファイルシステム (NFS) の概要のセクション。

*AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide* の NFS デーモンの始動方法、NFS デーモンの停止方法、セキュア NFS を使用したファイルシステムのエクスポート、セキュア NFS を使用したファイルシステムのマウントのセクション。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager* 管理ガイドの第 2 章 インストールおよびシステム要件のセクションを参照してください。

オペレーティング・システムおよびデバイスのマネージの System Management Interface Tool のセクション。



NIS+ のリファレンス。

---

## rm\_nisd デーモン

### 目的

**rpc.nisd** デーモンを停止し、**/etc/rc.nfs** ファイルのエントリーにコメントを付けます。

### 構文

```
rm_nisd [ -I | -B | -N ]
```

### 説明

**rm\_nisd** デーモンは、**/etc/rc.nfs** ファイルの **rpc.nisd** デーモンのエントリーにコメントを付けます。

**rm\_nisd** デーモンは、**stopsrc** コマンドを使って **rpc.nisd** デーモンを停止します。

注: **mk\_nisd**、**mk\_cachemgr**、**mk\_nispasswd**、**rm\_nisd**、**rm\_cachemgr**、および **rm\_nispasswd** コマンドは、次の 2 つのことを行います。

- **/etc/rc.nfs** のデーモン始動コールのエントリーを変更する。
- デーモン **src** エンティティのデフォルトの動作を変更する。

例えば、**rpc.nisd** デーモンが **-Y** フラグによって始動されるとすると、このことは、**rpc.nisd** デーモンを始動するために **/etc/rc.nfs** エントリーでは明示的に設定されません。その代わりに、**chssys** が実行されて、始動時にデーモンに追加されるデフォルト・オプションがあればそのオプションが入れられます。これらのオプションが存在するか確認するには、**lssrc -S -s subsystem** コマンドを使用して、デフォルト・オプションを表示させます。

### フラグ

- I** **/etc/rc.nfs** ファイルの **rpc.nisd** デーモンのエントリーにコメントを付けます。
- B** **/etc/rc.nfs** ファイルの **rpc.nisd** デーモンのエントリーにコメントを付け、**rpc.nisd** デーモンを停止します。このフラグがデフォルトです。
- N** **stopsrc** コマンドを使って **rpc.nisd** デーモンを停止します。このフラグは **/etc/rc.nfs** ファイルを変更しません。

### 例

**rpc.nisd** デーモンを始動する **/etc/rc.nfs** ファイルのエントリーにコメントを付けるには、次のように入力します。

```
rm_nisd -I
```

このコマンドは、現在実行中のデーモンは停止しません。

### ファイル

**/etc/rc.nfs** NFS と NIS デーモンの始動スクリプトが入っています。

## 関連情報

**smit** コマンド。

**rpc.nisd** デーモン。

*AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*の Network Information Service (NIS+) Overview for System Management のセクション。

*AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*のネットワーク・ファイルシステム (NFS) の概要のセクション。

ネットワークおよびコミュニケーションのマネージ の NFS デーモンの始動方法、NFS デーモンの停止方法のセクション。

セキュリティーのセキュア NFS を使用したファイルシステムのエクスポートおよびセキュア NFS を使用したファイルシステムのマウントのセクション。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

オペレーティング・システムおよびデバイスのマネージのSystem Management Interface Toolのセクション。

NIS+ のリファレンス。

---

## rm\_nispasswdd デーモン

### 目的

**rpc.nispasswdd** デーモンを停止し、**/etc/rc.nfs** ファイルのエントリーにコメントを付けます。

### 構文

**rm\_nispasswdd** [ **-I** | **-B** | **-N** ]

### 説明

**rm\_nispasswdd** デーモンは、**/etc/rc.nfs** ファイルの **rpc.nispasswdd** デーモンのエントリーにコメントを付けます。 **rm\_nispasswdd** デーモンは、**stopsrc** コマンドを使って **rpc.nispasswdd** デーモンを停止します。

注: **mk\_nisd**、**mk\_cachemgr**、**mk\_nispasswdd**、**rm\_nisd**、**rm\_cachemgr**、および **rm\_nispasswdd** コマンドは、次の 2 つのことを行います。

- **/etc/rc.nfs** のデーモン始動コールのエントリーを変更する。
- デーモン **src** エンティティーのデフォルトの動作を変更する。

例えば、**rpc.nisd** デーモンが **-Y** フラグによって始動されるとすると、このことは、**rpc.nisd** デーモンを始動するために **/etc/rc.nfs** エントリーでは明示的に設定されません。その代わりに、**chssys** が実行されて、始動時にデーモンに追加されるデフォルト・オプションがあればそのオプションが入れられます。これらのオプションが存在するか確認するには、**lssrc -S -s subsystem** コマンドを使用して、デフォルト・オプションを表示させます。

## フラグ

- I `/etc/rc.nfs` ファイルの `rpc.nispasswdd` デーモンのエントリーにコメントを付けます。
- B `/etc/rc.nfs` ファイルの `rpc.nispasswdd` デーモンのエントリーにコメントを付け、`rpc.nispasswdd` デーモンを停止します。このフラグがデフォルトです。
- N `stopsrc` コマンドを使って `rpc.nispasswdd` デーモンを停止します。このフラグは `/etc/rc.nfs` ファイルを変更しません。

## 例

`rpc.nispasswdd` デーモンを始動する `/etc/rc.nfs` ファイルのエントリーにコメントを付けるには、次のように入力します。

```
rm_nispasswdd -I
```

このコマンドは、現在実行中のデーモンは停止しません。

## ファイル

`/etc/rc.nfs` NFS と NIS デーモンの始動スクリプトが入っています。

## 関連情報

`smit` コマンド。

`rpc.nispasswdd` デーモン。

*AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide* の Network Information Service (NIS+) Overview for System Management のセクション。

ネットワークおよびコミュニケーションのマネージ のネットワーク・ファイルシステム (NFS) の概要のセクション。

ネットワークおよびコミュニケーションのマネージ の NFS デーモンの始動方法、NFS デーモンの停止方法のセクション。

セキュリティーのセキュア NFS を使用したファイルシステムのエクスポートおよびセキュア NFS を使用したファイルシステムのマウントのセクション。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド* の第 2 章 インストールおよびシステム要件のセクションを参照してください。

NIS+ のリファレンス。

---

## rmail コマンド

### 目的

基本ネットワーク・ユーティリティー (BNU) を使用して受信したリモート・メールを処理します。

## 構文

**rmail** *User*

## 説明

**rmail** コマンドは **uucp** コマンドを使用して受信した着信メールを解釈します。**bellmail** コマンドが生成したフォーマットの From ヘッダー行を以下のように単一行のフォーマットにします。

```
return-path!sender
```

**rmail** コマンドは処理済みのメールを **sendmail** コマンドに渡します。*User* パラメーターは、**sendmail** コマンドによって認識されたユーザーを指定しなければなりません。

## 関連情報

**bellmail** コマンド、 **sendmail** コマンド、 **uucp** コマンド。

ネットワークおよびコミュニケーションのマネージ のメール管理のセクション。

---

## rmaudrec コマンド

### 目的

監査ログのレコードを除去します。

### 構文

```
rmaudrec [-a | -n node_name1[,node_name2]...] [-S subsystem_name]  
-s selection_string [-h] [-V]
```

### 説明

**rmaudrec** コマンドは、監査ログ内のレコードを削除します。監査ログは、システムのエベレーションに関する情報を記録するための機能です。監査ログには、システムの通常オペレーションに関する情報のほか、失敗や他のエラーも含めることができます。監査ログは、他システム・アクティビティーに関係したエラーの関連性を伝達することにより、エラー・ログの機能性を増大させます。障害に関する詳細情報もすべて AIX エラー・ログに書き込まれます。

監査ログへのレコードの作成は、そのために用意されたサブシステムにより行われます。例えば、イベント応答サブシステムは、管理者が定義した条件をモニターするためにバックグラウンドで実行され、条件が真になると、1 つまたは複数のアクションを起動します。このサブシステムはバックグラウンドで実行されるため、発生したすべてのイベント・セットおよびイベントに対して取られたすべてのアクションの結果を、オペレーターまたは管理者が把握するのは困難です。イベント応答サブシステムはその活動を監査ログに記録するので、管理者は、イベント応答サブシステムの活動を他のサブシステムの活動と同様に容易に表示できます。さらに、レコードを明示的に除去しなければならないこともあり、これはこのコマンドを使用して行うことができます。

監査ログ内の各レコードは名前付きフィールドを持っています。各フィールドには、そのレコードに対応する状態に関する情報を提供する値が含まれています。例えば、**Time** という名前のフィールドには、状態が起こった時刻が表示されます。各レコードは、一組の共通フィールドと一組のサブシステム固有のフィールドを持っています。共通フィールドは、監査ログ内のすべてのレコードに含まれています。サブシステム固有のフィールドはレコードごとに異なっています。フィールドの名前が重要になるのは、サブシステム名と

一緒に使用するときだけです。なぜならば、フィールドの名前は必ずしもすべてのシステムにわたって固有でないからです。各レコードは、どのサブシステム固有のフィールドがレコードに含まれているかを定義し、かつ状態を記述するメッセージを生成するための書式制御文字列を定義するテンプレートから派生しています。書式制御文字列は、レコード・フィールドを挿入として使用することもあります。サブシステムは、通常、多くのテンプレートを持っています。

フィールド名を選択文字列内の変数として使用し、削除するレコードを選択することができます。選択文字列は、各レコードの参照フィールドを使用して、各レコードと突き合わされます。一致したレコードはすべて除去されます。選択文字列は、**-s** フラグを使用して指定します。

選択文字列は、フィールド名、定数、および演算子からなる式です。選択文字列の構文は、C プログラミング言語の式と非常によく似ています。選択文字列を指定する方法については、「*RSCT: Administration Guide*」を参照してください。

共通フィールド名は、以下のとおりです。

#### フィールド    説明

**Time**            このレコードが対応する状態が発生した時刻を示します。値は 64 ビットの整数で、Unix Epoch (00:00:00 GMT 01/01/1970) 以降のマイクロ秒数を表します。時刻をより見やすいフォーマットで指定するには、以下の定数を参照してください。

**Subsystem**      レコードを生成したサブシステムを示します。これは文字列です。

**Category**        レコードを生成したサブシステムが決めた、監査レコードに対応する、状態の重要性を示します。有効な値は、**0** (通知) および **1** (エラー) です。

#### **SequenceNumber**

レコードに割り当てられた固有な 64 ビットの整数を指定します。監査ログ内のレコードで、これと同じシーケンス番号を持つものはほかにありません。

**TemplateId**      同じ内容と書式制御文字列を持つレコードに割り当てられたサブシステム依存の ID を指定します。この値は 32 ビットの無符号整数です。

**NodeName**        レコードを入手したノードの名前を示します。このフィールド名は選択文字列で使用できません。

「*RSCT: Administration Guide*」で説明している式の定数のほかに、このコマンドでは日付および時刻に関する次の構文を使用できます。

#### **#mmdhhmmYYYY**

このフォーマットは、ここに示されているパターンに従って解釈される 10 進数文字のシーケンスからなっています。パターン内のフィールドは、左から右へ、*mm* = 月、*dd* = 日、*hh* = 時間、*mm* = 分、*YYYY* = 年です。例えば **#010523042002** は、2002 年 1 月 5 日午後 11 時 4 分を表します。フィールドは右から左へ省略することができます。存在しない場合、デフォルトとして、*year* = 現在の年、*minutes* = 0、*hour* = 0、*day* = 1、および *month* = 現在の月が使用されます。

#### **#-mmdhhmmYYYY**

このフォーマットは上記のフォーマットと似ていますが、現在の日時に関連しています。例えば、値 **#-0001** は 1 日前を示し、値 **#-010001** は 1 カ月と 1 時間前を示しています。フィールドは、右から省略していき、0 で置き換えることができます。

削除と見なされ、選択文字列と一致する監査レコードを、**-S** フラグを使用して特定のサブシステムに限定できます。このフラグを指定すると、共通フィールド名の他に、サブシステム固有のフィールド名も選択文字列に使用できます。

**-n** フラグを使用して、監査ログ・レコードを削除する対象のノードを、1 セットの特定ノードに限定できます。このフラグを指定すると、リストされたノードのセットだけに限定して検索されます。それ以外の場合、検索は、`CT_MANAGEMENT_SCOPE` 環境変数で決められた現行の管理有効範囲内に定義された、すべてのノードに実行されます。

まず、同じ **-s** および **-n** フラグ値を指定した **Isaudrec** コマンドを使用して、削除されるレコードをリストすることをお勧めします。これにより、意図したよりも多くのレコードと選択文字列を突き合わせる可能性を最小にできます。

## フラグ

**-a** ドメイン内のすべてのノードからレコードを除去することを指定します。 **-n** と **-a** の両方のフラグを省略すると、ローカル・ノードから取り出したレコードのみが除去されます。

**-n** *node\_name1* [*node\_name2*]...

検査の対象となり、指定された選択文字列と一致するなどの基準を満たす場合に、削除の対象となる、監査ログ・レコードを含むノードのリストを指定します。ノード・グループ名を指定することもでき、これはノード名のリストに展開されます。 **-n** と **-a** の両方のフラグを省略すると、ローカル・ノードから取り出したレコードのみが削除されます。

**-S** *subsystem\_name*

サブシステム名を指定します。このフラグがあると、*subsystem\_name* で示されたレコードだけが削除の対象になります。削除対象レコードは、**-s** フラグにより、さらに限定することができます。サブシステム名がスペースを含む場合は、単一引用符または二重引用符で囲む必要があります。

古いバージョンとの互換性を保つためには、**-a** および **-S** フラグが無指定の場合のみ、**-n** フラグを使用して、サブシステム名を指定できます。

**-s** *selection string*

選択文字列を指定します。この文字列は、監査ログ内の各レコードに照らして評価されます。評価がゼロ以外の結果 (**TRUE**) になった場合、そのレコードは監査ログから除去されます。選択文字列がスペースを含む場合は、単一引用符または二重引用符で囲む必要があります。選択文字列を指定する方法については、「*RSCT: Administration Guide*」を参照してください。

レコード内のフィールドの名前は式の中で使用できます。 **-S** フラグが無指定の場合、共通フィールドの名前のみを使用できます。共通フィールド名とそのデータ型のリストについては、**説明**のセクションを参照してください。 **-S** フラグを指定した場合、共通フィールド名の他に、指定したサブシステムのフィールド名もすべて使用できます。

このフラグが無指定の場合、監査ログからレコードは除去されません。

**-h** コマンドの使用ステートメント (使用法) を標準出力に書き込みます。

**-v** コマンドの詳細メッセージを標準エラーに書き込みます。

## パラメーター

*field\_name1* [*field\_name2*...]

表示する、監査ログ・レコード内の 1 つまたは複数のフィールドを指定します。コマンド行に指定したフィールド名の順序で表示されます。フィールド名を何も指定しないと、デフォルトにより **Time**、**Subsystem**、**Severity**、および **Message** が表示されます。管理有効範囲がローカルでない場合、デフォルトにより **NodeName** が最初の列に表示されます。これらのフィールドおよびその他のフィールドの詳細については、**説明**のセクションを参照してください。



## セキュリティ

**-S** フラグを省略時に監査ログからレコードを除去するには、ユーザーは、レコードを除去する各ノード上のターゲット・リソース・クラスへの書き込みアクセス権限を持つ必要があります。 **-S** フラグを指定した場合、ユーザーは、レコードを除去する各ノード上の、**-S** フラグで指定されたサブシステムに対応する監査ログ・リソースに、書き込みアクセス権限を持つ必要があります。

許可は、各ノード上に存在する RMC アクセス制御リスト (ACL) ファイルにより制御されます。

## 終了状況

- 0 コマンドが正常に実行されました。
- 1 RMC でエラーが起きました。
- 2 コマンド行インターフェース・スクリプトでエラーが起きました。
- 3 コマンド行に入力したフラグが誤りです。
- 4 コマンド行に入力したパラメーターが誤りです。
- 5 誤ったコマンド行入力により、エラーが起きました。

## 環境変数

### CT\_CONTACT

リソース・モニターおよび制御 (RMC) デーモンを使用したセッションを行うシステムを判別します。 CT\_CONTACT をホスト名または IP アドレスに設定すると、コマンドは指定されたホストの RMC デーモンと連絡します。 CT\_CONTACT を設定しないと、コマンドは、そのコマンドが実行されているローカル・システムの RMC デーモンと連絡します。 RMC デーモン・セッションのターゲットと管理有効範囲により、このコマンドにより影響されるリソース・クラスまたはリソースが決めます。

### CT\_IP\_AUTHENT

CT\_IP\_AUTHENT 環境変数が存在する場合、RMC デーモンは、IP ベース・ネットワーク認証を使用して、CT\_CONTACT 環境変数が設定されている IP アドレスで指定されたシステム上の RMC デーモンに連絡します。 CT\_IP\_AUTHENT は、CT\_CONTACT が IP アドレスに設定されている場合にのみ意味を持ちます。ドメイン・ネーム・システム (DNS) サービスには依存しません。

### CT\_MANAGEMENT\_SCOPE

RMC デーモンとのセッションに使用される管理有効範囲を決定します (**-a** および **-n** フラグと併用します)。管理有効範囲は、監査ログ・レコードを削除できる、可能なターゲット・ノードのセットを決定します。 **-a** および **-n** フラグが無指定の場合、Local 有効範囲が使用されます。これらのフラグのいずれかを指定すると、CT\_MANAGEMENT\_SCOPE を使用して、管理有効範囲が直接決定されます。有効な値は以下のとおりです。

- 0 *local* 有効範囲を指定します。
- 1 *local* 有効範囲を指定します。
- 2 *peer domain* (ピア・ドメイン) 有効範囲を指定します。
- 3 *management domain* (管理ドメイン) 有効範囲を指定します。

この環境変数を設定しないと、*local* 有効範囲が使用されます。



## 実装上の固有な条件

このコマンドは、AIX 用の Reliable Scalable Cluster Technology (RSCT) ファイルセットの一部です。

## 標準出力

**-h** フラグが指定されている場合は、このコマンドの使用状況の説明文が標準出力に書き込まれます。

## 標準エラー

**-v** フラグを指定してこのコマンドが正常に終了した場合は、削除されたレコードの数を示すメッセージが標準エラーに書き込まれます。

## 例

1. CT\_MANAGEMENT\_SCOPE 環境変数で定義された管理有効範囲内の、すべてのノード上の監査ログから、すべてのレコードを除去するには、次のように入力します。

```
rmaudrec -s "Time > 0"
```

または

```
rmaudrec -s "SequenceNumber >= 0"
```

2. CT\_MANAGEMENT\_SCOPE 環境変数で定義された管理有効範囲内の、すべてのノード上の、1 週間より前のレコードをすべて除去するには、次のように入力します。

```
rmaudrec -s "Time < #-0007"
```

3. **mynode** および **yournode** ノード上の、**abc** サブシステムにより作成された、1 日より前のレコードをすべて除去するには、次のように入力します。

```
rmaudrec -S abc -s "Time < #-0001" -n mynode,yournode
```

## 位置

`/usr/sbin/rsct/bin/rmaudrec`

## 関連情報

ブック: 次の事項の説明については、「*RSCT: Administration Guide*」を参照してください。

- 選択文字列の指定方法
- 式の定数の使用法

コマンド: **lsaudrec**

---

## rmC2admin コマンド

### 目的

分散 C2 システム・ホストの構成ファイルを除去します。

### 構文

```
rmC2admin [ -m ]
```

## 説明

**rmC2admin** コマンドは、分散 C2 システムのシンボリック・リンクを実際のファイルで置き換えます。ディレクトリー **/etc/data.shared** は除去されます。 **-m** フラグを使用すると、**hd10sec** ファイルシステムおよび **/etc/data.master** ディレクトリーも除去されます。このオプションは、C2 システム内の他のすべてのホストが管理ホストを別のシステムで置き換えた場合、または C2 構成ファイルも除去した場合、その後でのみ使用すべきです。

**/etc/inittab** 内のシステム初期化スクリプトのエントリーも除去され、このシステムをリブートすると、C2 モードに構成されていないシステムになります。

このコマンドをマルチユーザー・モードで実行すると、C2 システムからのユーザー定義は保存されません。このコマンドをシングルユーザー・モードで実行すると、C2 システムからのユーザー定義は除去され、root ユーザーのみが有効なユーザー ID になります。

このコマンドを実行した後は直ちにシステムをリブートし、変更を有効にする必要があります。

## フラグ

**-m** ホストは管理マスターとして構成されました。

## 終了状況

- 0 C2 システム管理ホスト情報は正常に除去されました。
- 1 システムは C2 モードで操作するように構成されていませんでした。
- 2 システムは C2 オプションを使用してインストールされていませんでした。
- 3 C2 システム管理ホスト情報の除去でエラーが起きました。
- 4 無効なコマンド行オプションが使用されました。

## ファイル

**/usr/sbin/rmC2admin** **rmC2admin** コマンドが入っています。

## 関連情報

**chC2admin** コマンド、**isC2host** コマンド、**lsC2admin** コマンド、**mkC2admin** コマンド。

---

## rmCCadmin コマンド

### 目的

分散 Common Criteria を使用可能にしたシステム・ホストの構成ファイルを除去します。

### 構文

**rmCCadmin** [ **-m** ]

## 説明

**rmCCadmin** コマンドは、Common Criteria 使用可能システムのシンボリック・リンクを実際のファイルで置き換えます。ディレクトリー **/etc/data.shared** は除去されます。**-m** フラグを使用すると、**hd10sec** ファイルシステムおよび **/etc/data.master** ディレクトリーも除去されます。このオプションは、Common Criteria 使用可能システム内の他のすべてのホストが管理ホストを別のシステムで置き換えた場合、または Common Criteria 使用可能構成ファイルも除去した場合、その後でのみ使用すべきです。

**/etc/inittab** 内のシステム初期化スクリプトのエントリーも除去され、このシステムをリブートすると、Common Criteria 使用可能モード用に構成されていないシステムになります。

このコマンドをマルチユーザー・モードで実行すると、Common Criteria 使用可能システムからのユーザー定義は保存されます。このコマンドをシングルユーザー・モードで実行すると、Common Criteria 使用可能システムからのユーザー定義は除去され、root ユーザーのみが有効なユーザー ID になります。

このコマンドを実行した後は直ちにシステムをリブートし、変更を有効にする必要があります。

## フラグ

**-m** ホストは管理マスターとして構成されました。

## 終了状況

- 0 Common Criteria 使用可能システム管理ホスト情報は、正常に除去されました。
- 1 システムは Common Criteria 使用可能モードで操作するように構成されていませんでした。
- 2 システムは Common Criteria 使用可能オプションを使用してインストールされていませんでした。
- 3 Common Criteria 使用可能システム管理ホスト情報の除去でエラーが起きました。
- 4 無効なコマンド行オプションが使用されました。

## ファイル

**/usr/sbin/rmCCadmin** **rmCCadmin** コマンドが入っています。

## 関連情報

**chCCadmin** コマンド、**isCChost** コマンド、**lsCCadmin** コマンド、**mkCCadmin** コマンド。

---

## rmcctrl コマンド

### 目的

リソース・モニターおよび制御 (RMC) サブシステムを管理します。

### 構文

```
rmcctrl { -a | -A | -d | -k | -K | -m {R | E | D} | -M {R | E | D} | -p | -P | -q | -Q | -s |  
-t n | -T | -u n | -U | -v n | -V | -w n | -W | -x | -X | -z | -h }
```

## 説明

**rmcctrl** コマンドは、リソース・モニターおよび制御 (RMC) サブシステムの操作を制御します。このサブシステムは、サブシステム名 **ctrmc** およびグループ名 **rsct** を持つシステム・リソース・コントローラー (SRC) によって制御されます。RMC サブシステム定義はサブシステム・オブジェクト・クラスに追加され、RSCT (Reliable Scalable Cluster Technology) のインストール時に開始されます。さらに、エントリーが **/etc/inittab** ファイルに作成され、システムのブート時に RMC サブシステムが自動的に開始されるようになります。

注: RMC サブシステムの停止と始動は、**stopsrc** および **startsrc** コマンドを使用して行うことができますが、**rmcctrl** コマンドを使ってこれらの機能を実行することをお勧めします。

## フラグ

- a RMC サブシステムをサブシステム・オブジェクト・クラスに追加し、エントリーを **/etc/inittab** ファイルの末尾に置きます。
- A RMC サブシステムを追加し、始動します。
- d RMC サブシステムをサブシステム・オブジェクト・クラスから削除し、RMC エントリーを **/etc/inittab** ファイルから除去します。
- k RMC サブシステムを停止します。
- K RMC サブシステムおよびすべてのリソース・マネージャーを停止します。
- m RMC サブシステム・クライアント・メッセージ・ポリシーを指定します。このポリシーは、コマンドが RMC サブシステムとは別のノードで実行された (すなわち、CT\_CONTACT 環境変数が設定されている) ときに、RMC サブシステムと「*RSCT for AIX 5L: Technical Reference*」にリストされているコマンドとの間で送信されるメッセージに適用されます。これらのメッセージは TCP/IP を使用して送信されます。

このフラグは、RSCT バージョン 2.3.1.0 以降でのみサポートされます。コマンドが RSCT の初期バージョンからのものである場合は、“Enabled” ポリシーを使用する必要があります。

- R** クライアント・メッセージ・ポリシーが “Required” であることを示します。“Required” は、メッセージ認証が使用できる (使用される) 場合にのみ接続がオープンされたままになることを意味します。
- E** クライアント・メッセージ・ポリシーが “Enabled” であることを示します。“Enabled” はデフォルトです。メッセージ認証は、接続の両方の側でサポートされている場合に使用されます。
- D** クライアント・メッセージ・ポリシーが “Disabled” であることを示します。“Disabled” は、メッセージ認証が使用されないことを意味します。
- M RMC サブシステム・デーモン・メッセージ・ポリシーを指定します。このポリシーは、管理ドメイン・クラスター内の RMC サブシステム・デーモン間で送信されるメッセージに適用されます。これらのメッセージは、ユーザー・データグラム・プロトコル (UDP) を使用して送信されます。  
このフラグは、RSCT リリース 2.4.1.0 以降でのみサポートされます。このフラグの指定時は、指定されたメッセージ・ポリシーが有効となるのは、RMC サブシステムの次回開始時です。
- R** デーモン・メッセージ・ポリシーが “Required” であることを示します。“Required” は、メッセージ認証が使用できる (使用される) 場合にのみ 2 つのデーモンが通信することを意味します。

- E** デーモン・メッセージ・ポリシーが "Enabled" であることを示します。"Enabled" はデフォルトです。送信側デーモンと受信側デーモンがメッセージ認証をサポートする場合に、メッセージ認証が使用されます。
- D** デーモン・メッセージ・ポリシーが "Disabled" であることを示します。"Disabled" は、メッセージ認証が使用されないことを意味します。メッセージ認証を使用不可にすると、クラスター内のすべてのノードが同じになるように構成されているわけではない場合は、機能が失われることがあります。
- p** リモート・クライアント接続を使用可能にします。
- P** リモート・クライアント接続を使用不可にします。
- q** RMC サブシステムが次に開始される時にリモート・クライアント接続を使用可能にします。
- Q** RMC サブシステムが次に開始される時にリモート・クライアント接続を使用不可にします。
- s** RMC サブシステムを開始します。
- t n** クライアント・メッセージのタイムアウト値を  $n$  秒に設定します。この時間内に、次のようにならないとなりません。
- RMC サブシステムがクライアント接続を受け入れた後で、開始セッション・プロトコルの最初のメッセージが到着する。
  - クライアント・メッセージの始まりの部分が受信されたら、そのメッセージの全体が RMC サブシステムによって受信される。
- これらの時間制限のいずれかを超えると、クライアント・セッションは閉じられます。最小許容値は **10**、最大は **86400** です。
- 指定された場合、この値は RMC サブシステムが次回に開始されたときに有効になります。
- T** クライアント・メッセージのタイムアウト値をデフォルト値 **10** 秒に設定します。
- 指定された場合、この値は RMC サブシステムが次回に開始されたときに有効になります。
- u n** 開始セッションのタイムアウト値を  $n$  秒に設定します。この時間内に、新しいクライアント・セッションのための開始セッション処理が完了しなければなりません。完了しない場合は、セッションは閉じられます。最小許容値は **60**、最大は **86400** です。
- 指定された場合、この値は RMC サブシステムが次回に開始されたときに有効になります。
- U** 開始セッションのタイムアウト値をデフォルト値 **300** 秒に設定します。
- 指定された場合、この値は RMC サブシステムが次回に開始されたときに有効になります。
- v n** 最初のコマンドのタイムアウト値を  $n$  秒に設定します。クライアント・セッションが RMC サブシステムで確立されたときに最初のコマンド・タイマーが設定された場合、最初のコマンドは、開始セッション処理の完了後、指定された秒数以内に到着しなければなりません。到着しない場合は、セッションが閉じられます。最小許容値は **10**、最大は **86400** です。
- 指定された場合、この値は RMC サブシステムが次回に開始されたときに有効になります。
- V** 最初のコマンドのタイムアウト値をデフォルト値 **10** 秒に設定します。
- 指定された場合、この値は RMC サブシステムが次回に開始されたときに有効になります。
- w n** 最初のコマンドのしきい値を  $n$  個のクライアント・セッションに設定します。クライアント・セッションの数がこの値を超えると、RMC サブシステムは、新しい非認証セッションごとに最初のコマンド・タイマーを使用可能にします。しきい値を **0** に設定した場合は、最初のコマンドのタイムアウト機能は使用不可になります。最大値は **150** です。

- 指定された場合、この値は RMC サブシステムが次回に開始されたときに有効になります。
- W** 最初のコマンドのしきい値をデフォルト値 **150** 個のクライアント・セッションに設定します。  
指定された場合、この値は RMC サブシステムが次回に開始されたときに有効になります。
  - x** 非 **root** 認証クライアント・セッションおよび非認証クライアント・セッションの最初のコマンド・タイムアウトを使用可能にします。  
指定された場合、この値は RMC サブシステムが次回に開始されたときに有効になります。
  - X** 非 **root** 認証セッションの最初のコマンド・タイムアウトを使用不可にします。  
指定された場合、この値は RMC サブシステムが次回に開始されたときに有効になります。
  - z** RMC サブシステムおよびすべてのリソース・マネージャーを停止しますが、RMC サブシステムおよびリソース・マネージャーが実際に停止されるまで、コマンドは戻りません。
  - h** コマンドの使用ステートメント (使用法) を標準出力に書き込みます。

## セキュリティ

特権制御: **root** ユーザーのみが、このコマンドの実行権 (**x**) を持つようにすべきです。

## 終了状況

- 0** コマンドが正常に実行されました。
- 1** コマンドは正常終了しませんでした。

## 標準出力

- h** フラグが指定されている場合は、このコマンドの使用状況の説明文が標準出力に書き込まれます。

## 例

1. RMC サブシステムを追加するには、次のように入力します。  
`rmcctrl -a`
2. RMC サブシステムを開始するには、次のように入力します。  
`rmcctrl -s`
3. RMC サブシステムを停止するには、次のように入力します。  
`rmcctrl -k`
4. RMC サブシステムを削除するには、次のように入力します。  
`rmcctrl -d`

## 位置

`/usr/sbin/rsct/bin/rmcctrl`

---

## rmcifscred コマンド

### 目的

`/etc/cifs_fs/cifscred` ファイルに保管された、特定のサーバーおよびユーザー・エントリーの CIFS 信用証明情報を除去します。

## 構文

```
rmcifscred -h RemoteHost -u user
```

## 説明

**rmcifscred** コマンドはサーバーおよびユーザー名を入力として受け取ります。この入力に **/etc/cifs\_fs/cifscred** にリストされた信用証明情報があると、それらの信用証明情報が除去されます。その後は、特定のユーザーが特定のサーバーにマウントする場合、パスワードの入力が必要となります。

## フラグ

**-h RemoteHost**

リモート・ホスト (CIFS サーバー) の名前を指定します。これをホスト名、IP アドレス、あるいは完全修飾ドメイン名として提供することができます。

**-u user**

**cifscred** ファイルから特定のサーバーに対する信用証明情報を除去したい対象のユーザー名を指定します。

## 終了状況

0  
>0

コマンドは正常に実行されました。  
エラーが発生しました。

## 例

1. server1 にマウントするために **/etc/cifs\_fs/cifscred** に保管された user1 の信用証明情報を除去するには、次のように入力します。

```
rmcifscred -h server1 -u user1
```

## 位置

**/usr/sbin/rmcifscred**

## ファイル

**/etc/cifs\_fs/cifscred**

CIFS 信用証明情報を保管します。

## 関連情報

chcifscred コマンド、chcifsmt コマンド、lscifscred コマンド、lscifsmt コマンド、mkcifscred コマンド、mkcifsmt コマンド、『rmcifsmt コマンド』。

---

## rmcifsmt コマンド

### 目的

CIFS マウントを**/etc/filesystems** ファイルから除去し、そのエントリーがマウントされていれば、それをアンマウントします。



## 構文

**rmcifsmt** -f *MountPoint* [-B | -N]

## 説明

**rmcifsmt** コマンドは、CIFS エントリーを **/etc/filesystems** から除去します。 エントリーがマウントされていれば、**rmcifsmt** コマンドはそれをアンマウントします。

## フラグ

|                             |                                                                               |
|-----------------------------|-------------------------------------------------------------------------------|
| <b>-B</b>                   | 対応するエントリーを <b>/etc/filesystems</b> ファイルから除去し、そのファイルシステムをアンマウントします。これはデフォルトです。 |
| <b>-f</b> <i>MountPoint</i> | CIFS マウントのパス名を指定します。                                                          |
| <b>-N</b>                   | ファイルシステムをアンマウントしますが、 <b>/etc/filesystems</b> ファイルからエントリーは除去しません。              |

## 終了状況

|    |                  |
|----|------------------|
| 0  | コマンドは正常に実行されました。 |
| >0 | エラーが発生しました。      |

## 例

1. **/mnt** にマウントされている CIFS マウントを除去し、それをアンマウントするには、次のように入力します。

```
rmcifsmt -f /mnt
```

## 位置

**/usr/sbin/rmcifsmt**

## ファイル

|                         |                   |
|-------------------------|-------------------|
| <b>/etc/filesystems</b> | CIFS エントリーを保管します。 |
|-------------------------|-------------------|

## 関連情報

chcifscred コマンド、chcifsmt コマンド、lscifscred コマンド、lscifsmt コマンド、mkcifscred コマンド、mkcifsmt コマンド、783 ページの『rmcifscred コマンド』。

---

## rmclass コマンド

### 目的

ワークロード・マネージメント・クラスを除去します。

## 構文

**rmclass** [ **-d** *Config\_Dir* ] [ **-S** *SuperClass* ] *Name*

## 説明

**rmclass** コマンドは、*Name* パラメーターによって識別されたスーパークラスまたはサブクラスを、クラス定義ファイル、クラス制限ファイル、およびクラス共有ファイルから除去します。このクラスは既存のクラスでなければなりません。事前定義の **Default** クラス、**System** クラス、および **Shared** クラスは除去できません。

さらに、**Super** スーパークラスを除去すると、**/etc/wlm/Config\_Dir/Super** ディレクトリーとそれに含まれているすべての WLM 属性ファイル (それが存在する場合) が除去されます。ユーザー作成のサブクラス (**Default** および **Shared** 以外のサブクラス) がまだ残っていると、スーパークラスの除去は失敗します。

注: スーパークラスを除去できるのは root のみです。スーパークラスの **adminuser** および **admingroup** 属性に指定されたユーザー名またはグループ名と一致するユーザー ID またはグループ ID を持つ root ユーザーまたは許可ユーザーのみが、このスーパークラスのサブクラスを除去できます。

通常、**rmclass** は、該当する WLM 属性ファイル内のクラスとその属性を削除し、この変更結果は、WLM の更新の後でのみ、**wlmcntrl** コマンドを使用して、メモリー内のクラス定義 (アクティブ・クラス) に適用されます。

空の文字列が **-d** フラグの付いた構成名 (*Config\_dir*) として渡された場合は、WLM メモリー内データ構造のクラスのみが更新され、属性ファイルは更新されません。したがって、クラスがまだ WLM 構成に定義されたままになっている場合は、WLM の更新または始動の後で再作成されます。このフラグは、主として、WLM API を使用してアプリケーションによってのみメモリー内 WLM データ構造内に動的に作成されたクラス (例えば、アプリケーション失敗の後で何らかのクリーンアップを行うために) を除去するために使用されます。

注: このコマンドは、時間ベースの構成のセット (**-d** フラグを使用してセットを指定しない) には適用できません。現行構成がセットの場合、**-d** フラグを指定して、どの通常構成にコマンドを適用すべきかを指定する必要があります。

## フラグ

**-d** *Config\_Dir* **/etc/wlm/Config\_dir** を属性ファイル用の代替ディレクトリーとして使用します。このフラグを使用しない場合、**/etc/wlm/current** によって指し示されたディレクトリー内の構成ファイルが使用されます。空の文字列が構成名 (**-d ""**) として渡された場合は、WLM メモリー内データ構造のみのクラスが削除され、構成ファイルは修正されません。

**-S** *SuperClass* サブクラスを除去するときにスーパークラスの名前を指定します。**Super** スーパークラスの **Sub** サブクラスを指定するには、次の 2 つの方法があります。

1. サブクラスのフルネームを **Super.Sub** として指定し、**-S** を使用しない。
2. **-S** フラグを指定してスーパークラス名を提供し、サブクラスの短縮名を使用する。

**rmclass options -S Super Sub**

## ファイル

|                |                    |
|----------------|--------------------|
| <b>classes</b> | クラスの名前と定義が含まれています。 |
| <b>limits</b>  | リソース限度が含まれています。    |
| <b>shares</b>  | リソース共有が含まれています。    |

## 関連情報

**wlmcntrl** コマンド、 **lsclass** コマンド、 **chclass** コマンド、 **mkclass** コマンド。

---

## rmcomg コマンド

### 目的

定義済みの通信グループをピア・ドメインから除去します。

### 構文

```
rmcomg [-q] [-h] [-TV] communication_group
```

### 説明

**rmcomg** コマンドは、オンライン・ピア・ドメインの、 *communication\_group* パラメーターで指定された名前を持つ、既存の通信グループの定義を除去します。通信グループは、トポロジー・サービスが使用するハートビート・リングを定義し、各ハートビート・リングのチューナブル・パラメーターを定義するために使用されます。通信グループは、ピア・ドメイン内でハートビートに使用できるデバイスを決定します。

**rmcomg** コマンドは、通信グループが定義されているピア・ドメイン内で現在オンラインになっているノードで実行する必要があります。ドメインから通信グループを除去するには、過半数のノードがオンラインである必要があります。

通信グループは、インターフェース・リソースにより参照されてはなりません。通信グループへのインターフェース・リソースによる参照を除去するには、 **chcomg** コマンドを使用してください。

### フラグ

- q** 抑止モードを指定します。通信グループが存在しない場合、コマンドはエラーを戻しません。
- h** コマンドの使用ステートメント (使用法) を標準出力に書き込みます。
- T** コマンドのトレース・メッセージを標準エラーに書き込みます。これはソフトウェア・サービス担当者のみが使用します。
- V** コマンドの詳細メッセージを標準出力に書き込みます。

### パラメーター

*communication\_group*

ピア・ドメインから除去する、定義された通信グループの名前を指定します。

## セキュリティ

**rmcomg** コマンドのユーザーは、**IBM.CommunicationGroup** リソース・クラスに書き込み権限が必要です。デフォルトでは、ピア・ドメイン内のどのノード上の **root** も、構成リソース・マネージャーを経由して、このリソース・クラスに読み取りおよび書き込み権限を持ちます。

## 終了状況

- 0 コマンドが正常に実行されました。
- 1 RMC でエラーが起きました。
- 2 コマンド行インターフェース・スクリプトでエラーが起きました。
- 3 コマンド行に入力したフラグが誤りです。
- 4 コマンド行に入力したパラメーターが誤りです。
- 5 誤ったコマンド行入力により、エラーが起きました。
- 6 通信グループが存在しません。

## 環境変数

### CT\_CONTACT

リソース・モニターおよび制御 (RMC) デーモンを使用したセッションを行うシステムを指定します。CT\_CONTACT をホスト名または IP アドレスに設定すると、コマンドは指定されたホストの RMC デーモンと連絡します。CT\_CONTACT を設定しないと、コマンドは、そのコマンドが実行されているローカル・システムの RMC デーモンと連絡します。RMC デーモン・セッションのターゲットと管理有効範囲により、処理されるリソース・クラスまたはリソースが決められます。

### CT\_IP\_AUTHENT

CT\_IP\_AUTHENT 環境変数が存在する場合、RMC デーモンは、IP ベース・ネットワーク認証を使用して、CT\_CONTACT 環境変数が設定されている IP アドレスで指定されたシステム上の RMC デーモンに連絡します。CT\_IP\_AUTHENT は、CT\_CONTACT が IP アドレスに設定されている場合にのみ意味を持ちます。ドメイン・ネーム・システム (DNS) サービスには依存しません。

## 制約事項

このコマンドは、通信グループを除去するピア・ドメインに定義され、オンラインになっているノードで実行する必要があります。

## 実装上の固有な条件

このコマンドは、AIX 用の Reliable Scalable Cluster Technology (RSCT) ファイルセットの一部です。

## 標準入力

**-f "-"** または **-F "-"** フラグが指定されているときは、このコマンドで 1 つ以上のノード名が標準入力から読み取られます。

## 標準出力

**-h** フラグが指定されている場合は、このコマンドの使用状況の説明文が標準出力に書き込まれます。詳細メッセージはすべて、標準出力に書き込まれます。

## 標準エラー

トレース・メッセージはすべて、標準エラーに書き込まれます。

## 例

この例で、**nodeA** は **AppIDomain** に定義され、オンラインになっています。ピア・ドメイン **AppIDomain** の通信グループ定義 **ComGrp1** を除去するには、このコマンドを **nodeA** で実行します。

```
rmcomg ComGrp1
```

## 位置

```
/usr/sbin/rsct/bin/rmcomg
```

## 関連情報

ブック: 「*RSCT: Administration Guide*」(ピア・ドメイン操作についての説明)

コマンド: **chcomg**、**lscomg**、**lsrpdomain**、**lsrpnnode**、**mkcomg**、**preprnode**

情報ファイル: **rmccli** (RMC 関連のコマンドの一般情報)

---

## rmcondition コマンド

### 目的

条件を除去します。

### 構文

```
rmcondition [-f] [-q] [-h] [-TV] condition[:node_name]
```

### 説明

**rmcondition** コマンドは、*condition* パラメーターで指定された条件を除去します。条件を除去するには、条件が既に存在するものでなければなりません。リンクされている応答があっても、条件を除去する必要がある場合は、**-f** フラグを使用して、条件および応答とのリンクを強制的に除去します。**-f** フラグを指定しない場合は、応答とのリンクが存在しても、条件は除去されません。このコマンドは応答を除去しません。

システム・ソフトウェアを適切に機能させるために特定の条件が必要な場合、その条件はロックされることがあります。ロックされた条件はアンロックされない限り変更または取り除くことはできません。

**rmcondition** コマンドに指定した条件がロックされた場合、その条件は除去されません。代わりに、条件がロックされたという通知のエラーが作成されます。条件をアンロックするには、**chcondition** コマンドの **-U** フラグを使用することができます。ただし、条件はシステム・ソフトウェアが適切に機能するために必須であり、通常ロックされているので、これをアンロックする際は注意が必要です。

### フラグ

- f** 応答にリンクされていても、条件を強制的に除去します。 応答とのリンクも条件同様に除去されますが、応答は除去されません。
- q** *condition* が存在しない場合、エラーを戻しません。
- h** コマンドの使用ステートメント (使用法) を標準出力に書き込みます。

- T コマンドのトレース・メッセージを標準エラーに書き込みます。これはソフトウェア・サービス担当者のみが使用します。
- V コマンドの詳細メッセージを標準出力に書き込みます。

## パラメーター

- condition* 除去する条件の名前を指定します。
- node\_name* 条件が定義されているノードを指定します。*node\_name* を指定しないと、ローカル・ノードが使用されます。*node\_name* は、CT\_MANAGEMENT\_SCOPE 環境変数により定義された有効範囲内のノードです。

## セキュリティ

**rmcondition** を実行するユーザーは、**IBM.Condition** リソース・クラスに書き込み権限が必要です。権限は、相手のシステムのアクセス制御リスト (ACL) ファイルに指定します。

## 終了状況

- 0 コマンドが正常に実行されました。
- 1 RMC でエラーが起きました。
- 2 コマンド行インターフェース・スクリプトでエラーが起きました。
- 3 コマンド行に入力したフラグが誤りです。
- 4 コマンド行に入力したパラメーターが誤りです。
- 5 誤ったコマンド行入力により、エラーが起きました。

## 環境変数

### CT\_CONTACT

リソース・モニターおよび制御 (RMC) デーモンを使用したセッションを行うシステムを指定します。CT\_CONTACT をホスト名または IP アドレスに設定すると、コマンドは指定されたホストの RMC デーモンと連絡します。CT\_CONTACT を設定しないと、コマンドは、そのコマンドが実行されているローカル・システムの RMC デーモンと連絡します。RMC デーモン・セッションのターゲットと管理有効範囲により、処理されるリソース・クラスまたはリソースが決められます。

### CT\_IP\_AUTHENT

CT\_IP\_AUTHENT 環境変数が存在する場合、RMC デーモンは、IP ベース・ネットワーク認証を使用して、CT\_CONTACT 環境変数が設定されている IP アドレスで指定されたシステム上の RMC デーモンに連絡します。CT\_IP\_AUTHENT は、CT\_CONTACT が IP アドレスに設定されている場合にのみ意味を持ちます。ドメイン・ネーム・システム (DNS) サービスには依存しません。

### CT\_MANAGEMENT\_SCOPE

イベント応答リソース・マネージャー (ERRM) のリソースの処理で、RMC デーモンとのセッションに使用する管理有効範囲を指定します。管理有効範囲は、リソースを処理できる、可能なターゲット・ノードのセットを決定します。有効な値は以下のとおりです。

- 0 *local* 有効範囲を指定します。
- 1 *local* 有効範囲を指定します。
- 2 *peer domain* (ピア・ドメイン) 有効範囲を指定します。

3 *management domain* (管理ドメイン) 有効範囲を指定します。

この環境変数を設定しないと、*local* 有効範囲が使用されます。

## 標準出力

**-h** フラグが指定されている場合は、このコマンドの使用状況の説明文が標準出力に書き込まれます。詳細メッセージはすべて、標準出力に書き込まれます。

## 標準エラー

トレース・メッセージはすべて、標準エラーに書き込まれます。

## 例

以下の例は、スタンドアロン・システムに適用されます。

1. "FileSystem space used" という名前の条件定義を除去するには、次のコマンドを実行します。

```
rmcondition "FileSystem space used"
```

2. "FileSystem space used" という名前の条件定義を、条件が応答にリンクされていても除去するには、次のコマンドを実行します。

```
rmcondition -f "FileSystem space used"
```

この例は、管理ドメインに適用されます。

1. この例では、現行ノードは管理サーバーです。管理ノード **nodeB** に定義されている、"nodeB FileSystem space used" という名前の条件定義を除去するには、次のコマンドを実行します。

```
rmcondition "FileSystem space used:nodeB"
```

この例は、ピア・ドメインに適用されます。

1. 管理ノード **nodeA** に定義されている、"nodeA FileSystem space used" という名前の条件定義を除去するには、ドメイン内の任意のノードから次のコマンドを実行します。

```
rmcondition "nodeA FileSystem space used:nodeA"
```

## 位置

**/usr/sbin/rsct/bin/rmcondition**

## 関連情報

コマンド: **chcondition**、**lscondition**、**lscondresp**、**mkcondition**

ファイル: **rmcli**

---

## rmcondresp コマンド

### 目的

1 つの条件と 1 つ以上の応答との間のリンクを削除します。



## 構文

条件と 1 つまたは複数の応答との間のリンクを削除する場合:

```
rmcondresp [-q] [-h] [-TV] condition[:node_name] [response [response...]]
```

1 つまたは複数の応答への、すべてのリンクを削除する場合:

```
rmcondresp [-q] -r [-h] [-TV] response1 [response2...][:node_name]
```

条件/応答の関連付けをロックまたはアンロックする場合:

```
rmcondresp {-U | -L} [-h] [-TV] condition[:node_name] response
```

## 説明

**rmcondresp** コマンドは、条件と 1 つまたは複数の応答の間のリンクを削除します。条件と応答の間のリンクは、*条件/応答の関連付け* と呼ばれます。条件が発生しても、応答はもう実行されません。コマンド・パラメーターが応答だけからなることを指定するには、**-r** フラグを使用します。これは、このような応答の条件への全リンクを削除します。条件が 1 つだけ指定されている場合は、条件に対するすべての応答へのリンクが除去されます。

システム・ソフトウェアが正しく動作するために特定の条件/応答の関連付けが必要な場合には、それをロックすることができます。ロックされた条件/応答の関連付けを、**rmcondresp** コマンドによって除去することはできません。**rmcondresp** コマンドに指定した条件/応答の関連付けがロックされている場合、それは除去されません。その代わりに、条件/応答の関連付けがロックされていることを通知するエラーが生成されます。条件/応答の関連付けをアンロックするには、**-U** フラグを使用できます。しかし、条件/応答の関連付けはシステム・ソフトウェアの正常な動作のために必須であり、通常はロックされているので、これをアンロックする際は十分な注意が必要です。

## フラグ

- q** *condition* または *response* が存在しない場合、エラーを戻しません。
- r** コマンド・パラメーターすべてが応答であることを示します。条件は指定されていません。このコマンドは、指定された応答にリンクされているすべての条件から、条件/応答の関連付けを除去します。
- h** コマンドの使用ステートメント (使用法) を標準出力に書き込みます。
- T** コマンドのトレース・メッセージを標準エラーに書き込みます。これはソフトウェア・サービス担当者のみが使用します。
- V** コマンドの詳細メッセージを標準出力に書き込みます。
- U** 条件/応答の関連付けをアンロックして、その始動、停止、除去ができるようにします。条件/応答の関連付けのロックはシステム・ソフトウェアが正しく動作するために必須であり、当然のことです。このことから、このリンクをアンロックする場合は、十分な注意が必要です。**-U** フラグを使用して条件/応答の関連付けをアンロックする場合、このコマンドでそれ以外の操作を実行することはできません。
- L** 条件/応答の関連付けをロックし、その始動、停止、除去ができないようにします。**-U** フラグを使用して条件/応答の関連付けをロックする場合、このコマンドでそれ以外の操作を実行することはできません。

## パラメーター

|                  |                                                                                                             |
|------------------|-------------------------------------------------------------------------------------------------------------|
| <i>condition</i> | 応答にリンクされた条件の名前を指定します。-r フラグを使用する場合以外は、必ず条件を最初に指定します。                                                        |
| <i>response</i>  | 1 つまたは複数の応答の名前を指定します。指定された条件への、指定された応答からのリンクが除去されます。                                                        |
| <i>node_name</i> | 条件が定義されているノードを指定します。-r フラグを使用した場合は、応答が定義されているノードです。node_name は、CT_MANAGEMENT_SCOPE 環境変数により定義された有効範囲内のノードです。 |

## セキュリティ

**rmcondresp** を実行するユーザーは、**IBM.Association** リソース・クラスに書き込み権限が必要です。権限は、相手のシステムのアクセス制御リスト (ACL) ファイルに指定します。

## 終了状況

- 0 コマンドが正常に実行されました。
- 1 RMC でエラーが起きました。
- 2 コマンド行インターフェース・スクリプトでエラーが起きました。
- 3 コマンド行に入力したフラグが誤りです。
- 4 コマンド行に入力したパラメーターが誤りです。
- 5 誤ったコマンド行入力により、エラーが起きました。

## 環境変数

### CT\_CONTACT

リソース・モニターおよび制御 (RMC) デーモンを使用したセッションを行うシステムを指定します。CT\_CONTACT をホスト名または IP アドレスに設定すると、コマンドは指定されたホストの RMC デーモンと連絡します。CT\_CONTACT を設定しないと、コマンドは、そのコマンドが実行されているローカル・システムの RMC デーモンと連絡します。RMC デーモン・セッションのターゲットと管理有効範囲により、処理されるリソース・クラスまたはリソースが決められます。

### CT\_IP\_AUTHENT

CT\_IP\_AUTHENT 環境変数が存在する場合、RMC デーモンは、IP ベース・ネットワーク認証を使用して、CT\_CONTACT 環境変数が設定されている IP アドレスで指定されたシステム上の RMC デーモンに連絡します。CT\_IP\_AUTHENT は、CT\_CONTACT が IP アドレスに設定されている場合にのみ意味を持ちます。ドメイン・ネーム・システム (DNS) サービスには依存しません。

### CT\_MANAGEMENT\_SCOPE

イベント応答リソース・マネージャー (ERRM) のリソースの処理で、RMC デーモンとのセッションに使用する管理有効範囲を指定します。管理有効範囲は、リソースを処理できる、可能なターゲット・ノードのセットを決定します。有効な値は以下のとおりです。

- 0 *local* 有効範囲を指定します。
- 1 *local* 有効範囲を指定します。
- 2 *peer domain* (ピア・ドメイン) 有効範囲を指定します。

### 3 *management domain* (管理ドメイン) 有効範囲を指定します。

この環境変数を設定しないと、*local* 有効範囲が使用されます。

## 標準出力

**-h** フラグが指定されている場合は、このコマンドの使用状況の説明文が標準出力に書き込まれます。詳細メッセージはすべて、標準出力に書き込まれます。

## 標準エラー

トレース・メッセージはすべて、標準エラーに書き込まれます。

## 例

以下の例は、スタンドアロン・システムに適用されます。

1. 条件 "FileSystem space used" と応答 "Broadcast event on-shift" の間のリンクを削除するには、次のコマンドを実行します。

```
rmcondresp "FileSystem space used" "Broadcast event on-shift"
```

2. 条件 "FileSystem space used" とその応答すべての間のリンクを削除するには、次のコマンドを実行します。

```
rmcondresp "FileSystem space used"
```

3. 条件 "FileSystem space used" と応答 "Broadcast event on-shift" および "E-mail root anytime" の間のリンクを削除するには、次のコマンドを実行します。

```
rmcondresp "FileSystem space used" "Broadcast event on-shift" "E-mail root anytime"
```

4. 応答 "Broadcast event on-shift" と、これを使用するすべての条件の間のリンクを削除するには、次のコマンドを実行します。

```
rmcondresp -r "Broadcast event on-shift"
```

以下の例は管理ドメインに適用されます。

1. 管理サーバー上の条件 "FileSystem space used" と応答 "Broadcast event on-shift" の間のリンクを削除するには、次のコマンドを管理サーバーで実行します。

```
rmcondresp "FileSystem space used" "Broadcast event on-shift"
```

2. 管理ノード **nodeB** 上の条件 "FileSystem space used" と、応答 "Broadcast event on-shift" および "E-mail root anytime" 間のリンクを削除するには、次のコマンドを管理サーバーで実行します。

```
rmcondresp "FileSystem space used":nodeB \  
"Broadcast event on-shift" "E-mail root anytime"
```

以下の例はピア・ドメインに適用されます。

1. ドメイン内の **nodeA** 上の条件 "FileSystem space used" と、応答 "Broadcast event on-shift" および "E-mail root anytime" の間のリンクを削除するには、ドメイン内の任意のノードから、次のコマンドを実行します。

```
rmcondresp "FileSystem space used":nodeA \  
"Broadcast event on-shift" "E-mail root anytime"
```

2. ドメイン内の **nodeA** 上のすべての条件と、応答 "Broadcast event on-shift" 間のリンクを削除するには、次のコマンドをドメイン内の任意のノードから実行します。

```
rmcondresp -r "Broadcast event on-shift":nodeA
```

## 位置

/usr/sbin/rsct/bin/rmcondresp

## 関連情報

コマンド: **lscondresp**、**mkcondition**、**mkcondresp**、**mkresponse**、**startcondresp**、**stopcondresp**

ファイル: **rmccli**

---

## rmcosi コマンド

### 目的

共通オペレーティング・システム・イメージ (COSI) を除去します。

### 構文

**rmcosi** [-f] [-v] *COSI*

### 説明

**rmcosi** コマンドは、**mkcosi** コマンドを使用して作成された共通オペレーティング・システム・イメージ (COSI) を除去します。除去対象のその共通イメージがシン・サーバーによって使用されている場合は、強制フラグ (-f) が指定されていない限り、その操作は失敗します。-f フラグはすべてのシン・サーバーの共通イメージのセッションを終了させるため、COSI を除去することができます。このコマンドは、システム上に存在する **bos.sysmgt.nim.master** ファイルセットに依存します。

### フラグ

|    |                                                                                         |
|----|-----------------------------------------------------------------------------------------|
| -f | 共通イメージの除去を強制実行します。共通イメージがシン・サーバーによって使用されている場合は、シン・サーバーはオフラインにされ、その結果、共通イメージの除去が可能となります。 |
| -v | <b>rmcosi</b> コマンドの実行時に詳細デバッグ出力を有効にします。                                                 |

### 終了状況

|    |                  |
|----|------------------|
| 0  | コマンドは正常に実行されました。 |
| >0 | エラーが発生しました。      |

### セキュリティ

アクセス制御: **rmcosi** コマンドを実行するには、root 権限が必要です。

### 例

1. **cosi1** という名前の共通イメージを除去するには、次のように入力します。

```
rmcosi cosi1
```

## 位置

/usr/sbin/rmcosi

## ファイル

/etc/niminfo

NIM が使用する変数が入っています。

## 関連情報

**chcosi** コマンド、**cpcosi** コマンド、**lscosi** コマンド、**mkcosi** コマンド、**mkts** コマンド、**nim** コマンド、**nim\_clients\_setup** コマンド、**nim\_master\_setup** コマンド、**nimconfig** コマンド。

---

## rmddel コマンド

### 目的

SCCS ファイルからデルタを除去します。

### 構文

```
rmddel -r SID File ...
```

### 説明

**rmddel** コマンドは、*File* パラメーターで指定された各ソース・コード制御システム (SCCS) から、*SID* 変数で指定されたデルタを除去します。分岐内で最後に作成されたデルタ、または分岐がない場合は最新のトランク・デルタしか除去できません。また、指定する *SID* は、デルタを作成するために現在編集中のバージョンであってはなりません。デルタを除去するには、SCCS ファイルとディレクトリーを所有しているか、除去したいデルタを作成したユーザーでなければなりません。

*File* パラメーターにディレクトリーを指定すると、**rmddel** コマンドは要求されたアクションをすべての SCCS ファイル (ファイル名の先頭が **s** のファイル) に対して取ります。*File* パラメーターに **-** (ハイフン) を指定すると、**rmddel** コマンドは標準入力を読み取って、各行を SCCS ファイル名として解釈します。**rmddel** コマンドは、ファイルの終わり文字に達するまで、入力の読み取りを続けます。

デルタは、除去されると、**get** コマンドが作成する *g* ファイル内に含まれなくなります。ただし、**s** ファイル内には引き続きデルタ・テーブル・エントリーが残り、エントリーにはデルタが除去されたことを示す **R** が付いています。

### フラグ

**-r SID** SCCS ファイルから指定のデルタ *SID* を除去します。このフラグは必ず指定します。

### 終了状況

このコマンドは、以下の終了値を戻します。

**0** 正常終了。  
**>0** エラーが発生しました。

## 例

デルタ 1.3 を **s.test.c** SCCS ファイルから除去するには、次のように入力します。

```
rmdev -r 1.3 s.test.c
```

## ファイル

**/usr/bin/rmdev**

**rmdev** コマンドが入っています。

**s.files**

**rmdev** コマンドが処理するファイル。

## 関連情報

**delta** コマンド、 **get** コマンド、 **prs** コマンド。

**sccsfile** ファイル・フォーマット。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」のソース・コード制御システム (SCCS) の概要のセクション。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の SCCS コマンドのリストのセクション。

---

## rmdev コマンド

### 目的

システムからデバイスを除去します。

### 構文

```
rmdev { -l | -p }Name [ -d | -S ] [ -f File ] [ -h ] [ -q ] [ -R ]
```

### 説明

注: **-l** フラグは、**-p** フラグが指定されている場合は指定できません。**-R** フラグと **-p** フラグを同時に指定した場合は、**-R** フラグが無視されます。

**rmdev** コマンドは、**-l Name** フラグを使用して、デバイス論理名で指定されたデバイスを構成解除するか、または構成解除して未定義にするという両方を行います。デフォルト・アクションでは、デバイスを構成解除しますが、カスタマイズ・デバイス・オブジェクト・クラス内のデバイス定義は残します。

**-S** フラグを指定すると、停止状態がサポートされるデバイスの場合は、**rmdev** コマンドはデバイスを停止状態に設定します。**-d** フラグを指定すると、**rmdev** コマンドはカスタマイズ・デバイス・オブジェクト・クラスからデバイス定義を削除します (未定義)。**-d** フラグを指定しなければ、**rmdev** コマンドはデバイスを定義済みの状態に設定します (構成解除)。**-R** フラグを指定すると、**rmdev** コマンドはそのデバイスの子デバイスにも機能します。

すべての子デバイスを構成解除または削除するには、**-p** フラグを親デバイスの論理名と一緒に使用します。子デバイスは、**-R** フラグで記述されたのと同じ再帰的方法で構成解除または削除されますが、指定されたデバイス自身は構成解除または削除されません。

**重要:** 構成データベースを保護するために、**rmdev** コマンドは割り込みができないようになっています。このコマンドを、完了しないうちに停止すると、データベースが破壊されます。

また、Web-based System Manager の デバイス・アプリケーション、あるいは System Management Interface Tool (SMIT) **smit rmdev** 高速パスを使用して、このコマンドを実行することもできます。

## フラグ

|                |                                                                                                                        |
|----------------|------------------------------------------------------------------------------------------------------------------------|
| <b>-d</b>      | カスタマイズ・デバイス・オブジェクト・クラスからデバイス定義を除去します。このフラグは <b>-S</b> フラグとともに使用できません。                                                  |
| <b>-f File</b> | <i>File</i> パラメーターから必要なフラグを読み取ります。                                                                                     |
| <b>-h</b>      | コマンド使用方法メッセージを表示します。                                                                                                   |
| <b>-l Name</b> | カスタマイズ・デバイス・オブジェクト・クラス内の、 <i>Name</i> パラメーターで示された論理デバイスを指定します。このフラグは <b>-p</b> フラグとともに使用することはできません。                    |
| <b>-p Name</b> | 除去する必要がある子デバイスを保持する、カスタマイズ・デバイス・オブジェクト・クラス内の親論理デバイス ( <i>Name</i> パラメーターで示される) を指定します。このフラグは <b>-l</b> フラグとともに使用できません。 |
| <b>-q</b>      | 標準出力ならびに標準エラーからのコマンド出力メッセージを抑制します。                                                                                     |
| <b>-R</b>      | デバイスおよびその子デバイスを構成解除します。 <b>-d</b> または <b>-S</b> フラグとともに使うと、子デバイスは未定義または停止状態になります。                                      |
| <b>-S</b>      | デバイスに停止メソッドがある場合は、停止メソッドを呼び出すだけでデバイスを使用不可にします。このフラグは <b>-d</b> フラグとともに使用できません。                                         |

## 終了状況

このコマンドは、以下の終了値を戻します。

|    |             |
|----|-------------|
| 0  | 正常終了。       |
| >0 | エラーが発生しました。 |

## セキュリティ

特権制御: このコマンドに対する実行 (x) アクセス権は、root ユーザーとシステム・グループのメンバーだけが持ちます。

監査イベントは次のとおりです。

| イベント                   | 情報     |
|------------------------|--------|
| <b>DEV_Stop</b>        | デバイス名。 |
| <b>DEV_Unconfigure</b> | デバイス名。 |
| <b>DEV_Remove</b>      | デバイス名。 |

## 例

1. カスタマイズ・デバイス・オブジェクト・クラス内にデバイス定義を保存したまま、CD-ROM デバイス `cd0` を構成解除するには、以下のように入力します。

```
rmdev -l cd0
```

システムは下記のようなメッセージを表示します。

```
cd0 defined
```



2. CD-ROM デバイス `cd0` の定義をカスタマイズ・デバイス・オブジェクト・クラスから除去するには、以下のように入力します。

```
rmdev -d -l cd0
```

システムは下記のようなメッセージを表示します。

```
cd0 deleted
```

3. カスタマイズ・デバイス・オブジェクト・クラス内にデバイス定義を保存したまま、`scsi1` SCSI アダプターおよびその子すべてを構成解除するには、以下のように入力します。

```
rmdev -R -l scsi1
```

システムは下記のようなメッセージを表示します。

```
rmt0 Defined  
hdisk1 Defined  
scsi1 Defined
```

4. カスタマイズ・デバイス・オブジェクト・クラス内にデバイス定義を残したまま、`scsi1` SCSI アダプターの子を構成解除するが、アダプター自体は構成解除しないようにするには、以下のように入力します。

```
rmdev -p scsi1
```

システムは下記のようなメッセージを表示します。

```
rmt0 Defined  
hdisk1 Defined
```

5. カスタマイズ・デバイス・オブジェクト・クラス内にデバイス定義を保存したまま、`pci1` PCI バスの子およびその下の他のすべてのデバイスを構成解除するには、以下のように入力します。

```
rmdev -p pci1
```

システムは下記のようなメッセージを表示します。

```
rmt0 Defined  
hdisk1 Defined  
scsi1 Defined  
ent0 Defined
```

## ファイル

`/usr/sbin/rmdev` `rmdev` コマンドが入っています。

## 関連情報

`chdev` コマンド、`lsattr` コマンド、`lsconn` コマンド、`lsdev` コマンド、`lsparent` コマンド、`mkdev` コマンド。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

---

## rmmdir コマンド

### 目的

ディレクトリーを除去します。

## 構文

**rmdir** [ **-p** ] *Directory* ...

## 説明

**rmdir** コマンドは、*Directory* パラメーターで指定されたディレクトリーをシステムから除去します。また、その親ディレクトリーに書き込み許可がなければなりません。ディレクトリーが空かどうかを調べるには **ls -al** コマンドを使います。このディレクトリーは、NFS バージョン 4 サーバー用にエクスポートしてはなりません。

注: **rmdir** コマンドは、**-** (ダッシュ、ダッシュ) パラメーターをフラグの終わりを示す区切り文字としてサポートします。

## フラグ

**-p***Directory*      *Directory* パラメーターで指定されたパス名に従ってすべてのディレクトリーを除去します。親ディレクトリーが空で、しかも親ディレクトリーに書き込み許可がないと、これらのディレクトリーを除去できません。

## 終了状況

このコマンドは、以下の終了値を戻します。

**0**      *Directory* パラメーターで指定された各ディレクトリー・エントリーが正常に除去されました。  
**>0**      エラーが発生しました。

## 例

1. ディレクトリーを空にして除去するには、以下のように入力します。

```
rm mydir/* mydir/*.*
rmdir mydir
```

コマンドは、**mydir** ファイルの内容を除去してから、空のディレクトリーを除去します。**rm** コマンドがディレクトリー **.** (ドット) と **..** (ドット、ドット) を除去しようとしたことに関するメッセージを表示すると、**rmdir** コマンドはそれらを除去します。

**rm mydir/\* mydir/\*.\*** は、ドットで始まらない名前を持つファイルをまず除去してから、ドットで始まる名前を持つファイルを除去することに注意してください。**-a** フラグを使わない限り、**ls** コマンドは通常、ドットで始まるファイル名をリストしないので、ディレクトリーにドットで始まるファイル名が入っているかどうかはわかりません。

2. **/home**、**/home/demo**、および **/home/demo/mydir** ディレクトリーを除去するには、次のように入力します。

```
rmdir -p /home/demo/mydir
```

このコマンドは、最初に **/mydir** ディレクトリーを除去してから、**/demo** ディレクトリーと **/home** ディレクトリーを別個に除去します。ディレクトリーを除去しようとしたときに、このディレクトリーが空でなかった場合あるいは書き込み許可がなかった場合、コマンドは終了します。

# ファイル

`/usr/bin/rmdir` `rmdir` コマンドが入っています。

## 関連情報

オペレーティング・システムおよびデバイスのマネージのファイルのセクション。

オペレーティング・システムおよびデバイスのマネージのディレクトリーのセクション。

オペレーティング・システムおよびデバイスのマネージのファイルとディレクトリーのアクセス・モードのセクション。

`ls` コマンド、 `mkdir` コマンド、 `rm` コマンド。

`rmdir` サブルーチン、 `unlink` サブルーチン。

---

## rmf コマンド

### 目的

フォルダーと、そのフォルダーに含まれるメッセージを除去します。

### 構文

`rmf` [ + *Folder* ] [ -**interactive** | -**nointeractive** ]

### 説明

`rmf` コマンドは指定されたフォルダー内のメッセージを削除してから、フォルダーを削除します。デフォルトでは、`rmf` コマンドはフォルダーを削除する前にユーザーの要求を確認します。フォルダーにメッセージ以外のファイルが入っていると、`rmf` コマンドはそれらのファイルを削除せずにエラーを報告します。

**注意:** `rmf` コマンドは、他のリンクを持たないメッセージを必ず削除します。

デフォルトでは、`rmf` コマンドは現行フォルダーを除去します。現行フォルダーが除去されると、`inbox` が現行フォルダーになります。`+Folder` フラグを指定しなかった場合に、`rmf` コマンドが現行フォルダーを見つけられなければ、このコマンドは `+inbox` フォルダーを除去する前に確認を求めます。

フォルダーへのアクセス権が読み取り専用の場合は、`rmf` コマンドはフォルダーまたはフォルダーに含まれるメッセージを削除しません。`rmf` コマンドは、専用シーケンスと現行メッセージ情報のみをプロファイルから削除します。

`rmf` コマンドはフォルダーを再帰的に削除しません。親フォルダーの除去を要求してサブフォルダーを除去することはできません。サブフォルダーを除去すると、その親フォルダーが現行フォルダーになります。

### フラグ

`+Folder` 除去するフォルダーを指定します。

- help** コマンド構文、使用可能なスイッチ (トグル)、およびバージョン情報をリストします。  
注: メッセージ・ハンドラー (MH) の場合は、このフラグの名前は完全につづらなければなりません。
- interactive** フォルダを除去する前に確認を要求します。+Folder フラグを指定しない場合は、これがデフォルトです。
- nointeractive** 確認を要求せずにフォルダとそのメッセージを除去します。これはデフォルトです。

## プロフィール・エントリー

下記のエントリーは、*UserMhDirectory/mh\_profile* ファイルに入力します。

Current-Folder: デフォルトの現行フォルダを設定します。  
Path: ユーザーの MH ディレクトリーを指定します。

## 例

注意: **rmf** コマンドは、他のリンクを持たないメッセージを必ず削除します。

1. 現行フォルダ **status** を除去するには、以下のように入力します。

```
rmf
```

システムは次のようなメッセージで応答します。

```
Remove folder "status"?
```

フォルダを除去したい場合は **yes** と入力します。システムは次のようなメッセージで応答します。

```
[+inbox now current]
```

2. **meetings** フォルダを非対話式に除去するには、以下のように入力します。

```
rmf +meetings
```

## ファイル

**\$HOME/mh\_profile** MH ユーザー・プロフィールを定義します。  
**/usr/bin/rmf** **rmf** コマンドが入っています。

## 関連情報

**rmm** コマンド。

**mh\_alias** ファイル・フォーマット、**mh\_profile** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ のメール・アプリケーションのセクション。

---

## rmfilt コマンド

### 目的

フィルター規則をフィルター・テーブルから除去します。

### 構文

```
rmfilt -v 4|6 -n fid | all [-f]
```

## 説明

**rmfilt** コマンドは、フィルター規則テーブルからフィルター規則を除去するのに使用します。このコマンドによるアクションは、**mkfilt** コマンドが実行されるまで、IP セキュリティー・サブシステムには影響しません。このコマンドの IPsec フィルター規則は、**genfilt** コマンド、IPsec smit (IP バージョン 4 または IP バージョン 6)、または仮想プライベート・ネットワーク・サブメニュー内の Web-based System Manager を使用して構成することができます。

**rmfilt** コマンドは、フィルター規則テーブルからフィルター規則を除去します。除去できるのは、手動フィルター規則のみです。

## フラグ

- f** 自動生成フィルター規則を強制的に除去します。**-f** フラグを **-n all** と同時に指定すると、IP バージョン 4 に関する規則番号 1 を除き、すべてのフィルター規則 (ユーザー定義のフィルター規則と自動生成フィルター規則) が除去されます。
- n** フィルター規則テーブルから除去したいフィルター規則の ID です。IP バージョン 4 の場合、値 **1** はこのフラグには無効です。この値は、予約済みフィルター規則です。**all** を指定すると、**-f** フラグが指定されるまでは、ユーザー定義のフィルター規則がすべて除去されます。
- v** 除去する必要のあるフィルター規則の IP バージョンを示します。値 **4** は IP バージョン 4 を指定し、値 **6** は IP バージョン 6 を指定します。

---

## rmfs コマンド

### 目的

ファイルシステムを除去します。

### 構文

```
rmfs [ -r | -i ] FileSystem
```

### 説明

**rmfs** コマンドはファイルシステムを除去します。ファイルシステムがジャーナル・ファイルシステム (JFS または JFS2) の場合、**rmfs** ファイルシステムは、ファイルシステムが存在する論理ボリュームおよび **/etc/filesystems** ファイル内の関連するスタンザの両方を除去します。ファイルシステムが JFS または JFS2 ファイルシステムでない場合、このコマンドは **/etc/filesystems** ファイル内の関連するスタンザだけを除去します。*FileSystem* パラメーターは、除去するファイルシステムを指定します。

Web-based System Manager (wsm) のファイルシステム・アプリケーションを使用してファイルシステム特性を変更できます。

また、System Management Interface Tool (SMIT) **smit rmfs** 高速パスを使用して実行することもできます。

## フラグ

- r** ファイルシステムのマウント・ポイントを除去します。
- i** ファイルシステムを除去する前に警告を表示し、ユーザーにプロンプトを出します。

## 終了状況

このコマンドは、以下の終了値を戻します。

|    |                         |
|----|-------------------------|
| 0  | ファイルシステムは正常に除去されました。    |
| >0 | ファイルシステムは正常に除去されませんでした。 |

## セキュリティ

アクセス制御: root ユーザーまたは **system** グループのメンバーのみがこのコマンドを実行できます。

## 例

**/test** ファイルシステムを除去するには、以下のように入力します。

```
rmfs /test
```

これにより、**/test** ファイルシステム、**/etc/filesystems** ファイル内のそのエントリ、および基本となる論理ボリュームを除去します。

## ファイル

**/etc/rmfs**

**rmfs** コマンドが入っています。

**/etc/filesystems**

既知のファイルシステムをリストし、その特性を定義します。

## 関連情報

**chfs** コマンド、**crfs** コマンド、**lsfs** コマンド、**mkfs** コマンド、**rmlv** コマンド。

オペレーティング・システムおよびデバイスのマネージのファイルシステムのセクション。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

オペレーティング・システムおよびデバイスのマネージのSystem Management Interface Toolのセクション。

---

## rmgroup コマンド

### 目的

グループを除去します。

### 構文

```
rmgroup [ -R load_module ] Name
```

## 説明

**rmgroup** コマンドは、*Name* パラメーターで指定したグループを除去します。このコマンドは、すべてのグループ属性も削除します。グループを除去するには、そのグループ名が既に存在していなければなりません。グループ・メンバーであるユーザーは、システムから除去されません。

グループがあるユーザーの 1 次グループである場合、**chuser** コマンドを使用してそのユーザーの 1 次グループを再定義しない限り、そのグループは除去できません。**chuser** コマンドは、*/etc/passwd* ファイルを変更します。root ユーザー、または、GroupAdmin 権限が与えられているユーザーだけが、管理グループや管理ユーザーとして入っているグループを除去することができます。

代替 Identification and Authentication (I&A) メカニズムを使用して作成されたグループの場合、**-R** フラグを使用して、使用する I&A ロード・モジュールを指定できます。ロード・モジュールは、*/usr/lib/security/methods.cfg* ファイルに定義されます。

Web-based System Manager (wsm) の ユーザー アプリケーションを使用してユーザー特性を変更できます。

また、System Management Interface Tool (SMIT) **smit rmgroup** 高速パスを使用して実行することもできます。

## フラグ

**-R load\_module**           グループの除去に使用するロード可能 I&A モジュールを指定します。

## 終了状況

このコマンドは、以下の終了値を戻します。

**0**                            コマンドが正常に実行され、要求された変更がすべて行われました。  
**>0**                          エラーが発生しました。プリントされたエラー・メッセージに、障害のタイプについての詳細情報が提供されています。

## セキュリティ

アクセス制御: このコマンドは、実行 (x) アクセス権を root ユーザーとセキュリティ・グループのメンバーだけに与えます。このコマンドは、トラステッド・コンピューティング・ベース (TCB) 内のプログラムとしてインストールします。このコマンドは、**setuid** (SUID) ビットが設定されている root ユーザーが所有しなければなりません。

アクセスされるファイルは次のとおりです。

| モード       | ファイル                       |
|-----------|----------------------------|
| <b>r</b>  | <i>/etc/passwd</i>         |
| <b>rw</b> | <i>/etc/group</i>          |
| <b>rw</b> | <i>/etc/security/group</i> |

監査イベントは次のとおりです。

| イベント                | 情報   |
|---------------------|------|
| <b>GROUP_Remove</b> | グループ |



## 制約事項

グループの除去は、必ずしもすべてのロード可能 I&A モジュールによってサポートされているわけではありません。ロード可能 I&A モジュールがグループの除去をサポートしていない場合は、エラーが報告されます。

## 例

1. finance グループを除去するには、以下のように入力します。

```
rmgroup finance
```

2. LDAP I&A ロード可能なモジュール・グループを除去するには、以下のように入力します。

```
rmgroup -R LDAP monsters
```

## ファイル

`/usr/sbin/rmgroup`  
`/etc/group`  
`/etc/security/group`

`rmgroup` コマンドが入っています。  
グループの基本属性が入っています。  
グループの拡張属性が入っています。

## 関連情報

`chfn` コマンド、`chgrpmem` コマンド、`chsh` コマンド、`chgroup` コマンド、`chuser` コマンド、`lsgroup` コマンド、`lsuser` コマンド、`mkggroup` コマンド、`mkuser` コマンド、`passwd` コマンド、`pwdadm` コマンド、`rmuser` コマンド、`setgroups` コマンド、`setsenv` コマンド。

セキュリティーのネットワークの保護のセクション。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

---

## rmiscsi コマンド

### 目的

iSCSI ターゲット・データを削除します。

### 構文

```
rmiscsi -l AdapterName [ -g group ] [ -t TargetName ] [ -n PortNumber ] [ -i IPAddress ]
```

### 説明

`rmiscsi` コマンドは iSCSI ターゲット・データを ODM へ移します。ODM に格納されているデータには 2 つのカテゴリがあります。最初のカテゴリは、静的に構成された iSCSI ターゲットに対するもので、すべての関連する iSCSI ターゲット情報 (ターゲット名、IP アドレス、ポート番号など) を指定して、AIX がそれらを発見できるようにすることを要求します。iSCSI ターゲット・データの 2 番目のカテゴリは、動的に構成することができるが、ホストからの認証 (パスワードなど) を必要とする、iSCSI ターゲット・デバイス用のものです。iSCSI ターゲット・データのこれらの 2 つのカテゴリは、**-g** フラグにより指定される、**static** および **auto** グループにそれぞれ関係しています。

## フラグ

**-g** *group*

この iSCSI ターゲットがどのグループに関係するかを指定します。有効な 2 つのグループは、**static** および **auto** です。**static** グループは、このホストからは自動的に発見できない iSCSI ターゲット用であり、それらのすべての関連する iSCSI ターゲット情報 (ターゲット名、IP アドレス、ポート番号など) は指定する必要があります。**auto** グループは、自動的に発見されるが、認証情報 (パスワードなど) を必要とする、iSCSI ターゲット用のものです。

**-i** *IPaddress*

iSCSI ターゲットの IP アドレスを指定します。

**-l** *AdapterName*

この iSCSI ターゲットに接続される iSCSI TCP/IP

Offload Engine (TOE) アダプターのアダプター名を指定します。さらに、iSCSI ソフトウェア・ソリューション・デバイス用の iSCSI プロトコル・デバイスを指定することもできます。

**-n** *PortNumber*

iSCSI ターゲットがアクセスされるポート番号を指定します。デフォルトのポート番号は 3260 です。

**-t** *TargetName*

iSCSI ターゲット名 (例えば、iqn.sn9216.iscsi-hw1) を指定します。

## 終了状況

0

コマンドは正常に実行されました。

>0

エラーが発生しました。

## セキュリティ

**rmiscsi** コマンドを実行できるのは root のみです。

## 例

1. 1 つの静的構成された iSCSI ターゲットを除去するには、次のように入力します。

```
rmiscsi -l ics0 -g static -t iqn.sn1234.iscsi_hw1 -i 10.2.1.4 -n 3260
```

2. iSCSI TOE アダプター ics0 のすべての iSCSI ターゲットを除去するには、次のように入力します。

```
rmiscsi -l ics0
```

## 位置

**/usr/sbin/rmiscsi**

## ファイル

**src/bos/usr/sbin/iscsia**

iSCSI コマンドが構築された元となる共通ソース・ファイルを含んでいます。

## 関連情報

chiscsi コマンド、lscsi コマンド、mkiscsi コマンド。

---

## rmitab コマンド

### 目的

**/etc/inittab** ファイル内のレコードを除去します。

### 構文

**rmitab** *Identifier*

### 説明

**rmitab** コマンドは、**/etc/inittab** レコードを除去します。除去したいレコードを指定するには *Identifier* パラメーターを使用します。*Identifier* パラメーターには、オブジェクトを固有に識別するために使用される 1 文字から 14 文字までのフィールドを指定します。*Identifier* フィールドが固有でなければ、コマンドは失敗します。

### 例

tty2 用の tty エントリーを除去するには、以下のように入力します。

```
rmitab "tty002"
```

### 関連情報

**smit** コマンド、**chitab** コマンド、**lsitab** コマンド、**mkitab** コマンド、**init** コマンド、**/etc/inittab** ファイル。

---

## rmkeyerv コマンド

### 目的

**keyerv** デーモンを停止し、**/etc/rc.nfs** ファイルのエントリーにコメントを付けます。

### 構文

**/usr/sbin/rmkeyerv** [ **-I** | **-B** | **-N** ]

### 説明

**rmkeyerv** コマンドは、**/etc/rc.nfs** ファイルの **keyerv** デーモンに関するエントリーにコメントを付けます。**rmkeyerv** デーモンは、**stopsrc** コマンドを使用して **keyerv** デーモンを停止します。

Web-based System Manager (wsm) のファイルシステム・アプリケーションを使用してファイルシステム特性を変更できます。

また、System Management Interface Tool (SMIT) **smit rmkeyerv** 高速パスを使用して実行することもできます。

### フラグ

**-I** **/etc/rc.nfs** ファイルの **keyerv** デーモンに関するエントリーにコメントを付けます。

- B `/etc/rc.nfs` の `key serv` デーモンに関するエントリーにコメントを付け、`key serv` デーモンを停止します。このフラグがデフォルトです。
- N `stopsrc` コマンドを使用して `key serv` デーモンを停止します。このフラグは `/etc/rc.nfs` ファイルを変更しません。

## 例

`key serv` デーモンを始動させる `/etc/rc.nfs` ファイル内のエントリーにコメントを付けるには、以下のように入力します。

```
rmkeyserv -I
```

このコマンドは、現在実行中のデーモンは停止しません。

## ファイル

`/etc/rc.nfs` NFS と NIS デーモンの始動スクリプトが入っています。

## 関連情報

`smit` コマンド。

`key serv` デーモン。

ネットワークおよびコミュニケーションのマネージ のネットワーク・ファイルシステム (NFS) の概要のセクション。

ネットワークおよびコミュニケーションのマネージ の NFS デーモンの始動方法、NFS デーモンの停止方法のセクション。

セキュリティーのセキュア NFS を使用したファイルシステムのエクスポートおよびセキュア NFS を使用したファイルシステムのマウントのセクション。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

*AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*の Network Information Service (NIS) のセクション。

NIS のリファレンス。

---

## rmlpcmd コマンド

### 目的

1 つ以上の最小特権 (LP) リソースを Resource Monitoring and Control (RMC) サブシステムから除去します。

### 構文

1 つ以上の LP リソースを除去するには、以下のように入力します。

- ローカル・ノードの場合:

**rmlpcmd** [-h] [-TV] *resource\_name1* [ , *resource\_name2* , ... ]

- ドメイン内のすべてのノードの場合:

**rmlpcmd -a** [-h] [-TV] *resource\_name1* [ , *resource\_name2* , ... ]

- ドメイン内の一部のノードの場合:

**rmlpcmd -n** *host1* [,*host2*,...] [-h] [-TV] *resource\_name1* [ , *resource\_name2* , ... ]

## 説明

**rmlpcmd** コマンドは、1 つ以上の LP リソースを RMC サブシステムから除去します。LP リソースは、**root** のみが使用できるコマンドまたはスクリプトであり、これらのコマンドまたはスクリプトへのアクセスは、LP のアクセス制御リスト (ACL) 内のアクセス権に基づいてユーザーに権限付与されます。

**rmlpcmd** コマンドを使用して、ドメイン内の特定のノードまたはすべてのノードから、LP リソースを除去できます。ロックされた LP リソースを除去したい場合は、まず **chlpcmd** コマンドを使用してリソースの **Lock** 属性を設定解除する必要があります。

このコマンドはどのノードでも実行できます。ドメイン内のすべてのノードでこのコマンドを実行したい場合には、**-a** フラグを使用します。ドメイン内の一部のノードでこのコマンドを実行したい場合には、**-n** フラグを使用します。それ以外の場合、このコマンドはローカル・ノードで実行されます。

## フラグ

- a** 1 つ以上の LP リソースをドメイン内のすべてのノードから除去します。

**CT\_MANAGEMENT\_SCOPE** 環境変数の設定値によって、クラスター有効範囲が決まります。

**CT\_MANAGEMENT\_SCOPE** 変数が設定されていない場合、LP リソース・マネージャーは次の順序で有効範囲設定値を使用します。

- 管理ドメイン (あれば)
- ピア・ドメイン (あれば)
- ローカル有効範囲

**rmlpcmd** コマンドは、LP リソース・マネージャーが最初に検出した有効範囲に対して 1 回実行されます。例えば、管理ドメインとピア・ドメインが存在していて、かつ、その

**CT\_MANAGEMENT\_SCOPE** 環境変数が設定されていないと仮定します。この場合、**rmlpcmd -a** は管理ドメインで実行されます。 **rmlpcmd -a** をピア・ドメインで実行するためには、

**CT\_MANAGEMENT\_SCOPE** を **2** に設定する必要があります。

- n** *host1*[,*host2*,...]

LP リソースを除去するドメインの中の 1 つ以上のノードを指定します。デフォルトでは、LP リソースは、ローカル・ノードから除去されます。 **-n** フラグが有効なのは、管理ドメインまたはピア・ドメインの中だけです。 **CT\_MANAGEMENT\_SCOPE** 変数が設定されていない場合、LP リソース・マネージャーは次の順序で有効範囲設定値を使用します。

- 管理ドメイン (あれば)
- ピア・ドメイン (あれば)
- ローカル有効範囲

**rmlpcmd** コマンドは、LP リソース・マネージャーが最初に検出した有効範囲に対して 1 回実行されます。

- h** コマンドの使用ステートメント (使用法) を標準出力に書き込みます。

- T** コマンドのトレース・メッセージを標準エラーに書き込みます。

-V コマンドの詳細メッセージを標準出力に書き込みます。

## パラメーター

*resource\_name1[,resource\_name2,...]*

除去する 1 つ以上の LP リソースを指定します。

## セキュリティ

**rmlpcmd** コマンドを実行するには、**IBM.LPCCommands** リソース・クラスの Class ACL の中の読み取りおよび書き込み許可が必要です。アクセス権は、連絡先システムの LP ACL で指定されています。LP ACL に関する一般情報については **lpac1** 情報ファイル、LP ACL の変更については「*RSCT: Administration Guide*」を参照してください。

## 終了状況

- 0 コマンドが正常に実行されました。
- 1 RMC でエラーが起きました。
- 2 コマンド行インターフェース (CLI) スクリプトでエラーが発生しました。
- 3 コマンド行に間違ったフラグが指定されました。
- 4 コマンド行に不正なパラメーターが指定されました。
- 5 不正なコマンド行入力に基づく RMC により、エラーが発生しました。
- 6 リソースが見つかりません。

## 環境変数

### CT\_CONTACT

RMC デーモンとのセッションに使用されるシステムを決めます。 **CT\_CONTACT** にホスト名または IP アドレスが設定されていると、このコマンドは指定されたホスト上の RMC デーモンにアクセスします。 **CT\_CONTACT** が設定されていない場合、このコマンドは、コマンドを実行しようとするローカル・システムの RMC デーモンにアクセスします。 RMC デーモン・セッションのターゲットおよびその管理有効範囲によって、処理される LP リソースが決まります。

### CT\_MANAGEMENT\_SCOPE

LP リソースを処理するために RMC デーモンとのセッションに使用される管理有効範囲を決定します。管理有効範囲は、リソースの処理を行う場所となりうるターゲット・ノードのセットを決定します。有効な値は以下のとおりです。

- 0 *local* 有効範囲を指定します。
- 1 *local* 有効範囲を指定します。
- 2 *peer domain* (ピア・ドメイン) 有効範囲を指定します。
- 3 *management domain* (管理ドメイン) 有効範囲を指定します。

この環境変数を設定しないと、ローカルの 有効範囲が使用されます。

## 実装上の固有な条件

このコマンドは、AIX 用の Reliable Scalable Cluster Technology (RSCT) ファイルセットの一部です。

## 標準出力

**-h** フラグが指定されている場合は、このコマンドの使用状況の説明文が標準出力に書き込まれます。**-v** フラグが指定されると、このコマンドの詳細メッセージが標準出力に書き込まれます。

## 標準エラー

トレース・メッセージはすべて、標準エラーに書き込まれます。

## 例

1. **LP1** という名前の LP リソースを除去するには、次のように入力します。

```
rmllpcmd LP1
```

2. LP リソース **LP1** および **LP2** を除去するには、次のように入力します。

```
rmllpcmd LP1 LP2
```

## 位置

`/usr/sbin/rsct/bin/rmllpcmd` **rmllpcmd** コマンドが入っています。

## 関連情報

ブック: LP ACL の変更については、「*RSCT Administration Guide*」を参照してください。

コマンド: **chllpcmd**、**lphistory**、**lsllpcmd**、**mkllpcmd**、**runllpcmd**

情報ファイル: **lpacl** (LP ACL に関する一般情報)

---

## rmlv コマンド

### 目的

ボリューム・グループから論理ボリュームを除去します。

### 構文

```
rmlv [ -B ] [ -f ] [ -p Physical Volume] LogicalVolume ...
```

### 説明

**重要:** このコマンドは、指定した論理ボリュームに入っているすべてのデータを破壊します。

**rmlv** コマンドは論理ボリュームを除去します。LogicalVolume パラメーターは、論理ボリューム名または論理ボリューム ID のいずれかです。論理ボリュームは、まず最初にクローズしなければなりません。

*volume group* が並行モードで有効でない場合、*volume group* が有効である並行ノードのすべてで論理ボリュームはクローズされていなければなりません。例えば、論理ボリュームにファイルシステムが入っている場合は、マウントを解除しなければなりません。ただし、論理ボリュームを除去しても、そのボリューム上に存在していたファイルシステムが破壊されたことはオペレーティング・システムに通知されません。

**rmfs** コマンドは `/etc/filesystems` ファイルを更新します。

注:

1. このコマンドを使用するには、root ユーザー権を持っているか、または **system** グループのメンバーでなければなりません。



2. **rmlv** コマンドは、スナップショット・ボリューム・グループ、またはスナップショット・ボリューム・グループを持つボリューム・グループには使用できません。

Web-based System Manager (wsm) の ボリューム アプリケーションを使用してボリューム特性を変更できます。

また、System Management Interface Tool (SMIT) **smit rmlv** 高速パスを使って実行することもできます。

## フラグ

- B** 論理ボリュームが **-I** フラグを使用して作成された場合、親論理ボリュームに **chlvcopy -B -s** を発行します。それが正規の論理ボリュームである場合、**-B** フラグは無視されます。
- f** 確認を要求せずに論理ボリュームを除去します。
- p *PhysicalVolume*** *PhysicalVolume* 上の論理区画だけを除去します。他に未割り当ての物理区画がない限り、論理ボリュームは除去されません。
- 重要:** 論理ボリュームが複数の物理ボリュームにまたがっている場合は、*PhysicalVolume* 上の論理区画を除去するだけで、論理ボリューム全体の整合性が損なわれることがあります。

## 例

**注意:** この例で使用されているコマンドは、論理ボリュームに入っているデータをすべて破壊します。

ユーザーの確認を要求させずに論理ボリューム `lv05` を除去するには、以下のコマンドを入力します。

```
rmlv -f lv05
```

論理ボリュームがボリューム・グループから除去されます。

## ファイル

- |                         |                               |
|-------------------------|-------------------------------|
| <b>/usr/sbin/rmlv</b>   | <b>rmlv</b> コマンドが常駐するディレクトリー。 |
| <b>/tmp</b>             | コマンドの実行中に一時ファイルが格納されるディレクトリー。 |
| <b>/etc/filesystems</b> | 既知のファイルシステムをリストし、その特性を定義します。  |

## 関連情報

**varyonvg** コマンド、 **umount** コマンド、 **rmfs** コマンド。

オペレーティング・システムおよびデバイスのマネージの論理ボリューム・ストレージのセクションで、論理ボリューム・マネージャー、物理ボリューム、論理ボリューム、ボリューム・グループ、編成、データ保全性の確認、および割り当て特性について説明しています。

オペレーティング・システムおよびデバイスのマネージのSystem Management Interface Toolのセクションで、構造、メインメニュー、および SMIT で行なわれるタスクについて説明されています。

Web-based System Manager のインストールについては、 *AIX 5L Version 5.3 Web-based System Manager 管理ガイド* の第 2 章 インストールおよびシステム要件 のセクションを参照してください。

---

## rmlvcopy コマンド

### 目的

論理ボリュームからコピーを除去します。

### 構文

**rmlvcopy** *LogicalVolume Copies* [ *PhysicalVolume...* ]

### 説明

**rmlvcopy** コマンドは、*LogicalVolume* 内の各論理区画からコピーを除去します。コピーとは物理区画のことで、これらはオリジナルの物理区画とともに論理区画を構成します。1 つの論理ボリューム内に 2 つまでコピーを持つことができます。*Copies* パラメーターは、残す物理区画の最大数を決定します。

*LogicalVolume* パラメーターは、論理ボリューム名または論理ボリューム ID のいずれかです。

*PhysicalVolume* パラメーターは、物理ボリューム名または物理ボリューム ID のいずれかです。

*PhysicalVolume* パラメーターを使うと、その物理ボリュームからコピーのみが除去されます。

Web-based System Manager (wsm) の ボリューム アプリケーションを使用してボリューム特性を変更できます。

また、System Management Interface Tool (SMIT) **smit rmlvcopy** 高速パスを使って実行することもできます。

#### 注:

1. このコマンドを使用するには、root ユーザー権を持っているか、または **system** グループのメンバーでなければなりません。
2. ディスクが失敗したことを LVM が認識しなかった場合、LVM は別のミラーを除去する可能性があります。したがって、ディスクが失敗したことがわかっており、これらのディスクが存在しないことを LVM が表示しない場合は、コマンド行に失敗したディスクを指定するか、または **replacepv** を使用してディスクを置き換えるか、または **reducevg** を使用してディスクを除去する必要があります。
3. スナップショット・ボリューム・グループには **rmlvcopy** コマンドは使用できません。
4. **rmlvcopy** コマンドは、物理ボリューム名がコマンド行で指定され、指定された物理ボリュームがスナップショット・ボリューム・グループに属する場合にのみ、スナップショット・ボリューム・グループを持つボリューム・グループに使用できます。

### 例

論理ボリューム lv0112 に属する各論理区画のコピー数を減らすには、以下のように入力します。

```
rmlvcopy lv0112 2
```

論理ボリューム内の各論理区画は最高 2 つの物理区画を持つようになります。

### ファイル

**/usr/sbin/rmlvcopy**  
**/tmp/\***

**rmlvcopy** コマンドが入っています。  
コマンドの実行中に一時ファイルが格納されるディレクトリー。

## 関連情報

**mklv** コマンド、 **mklvcopy** コマンド。

オペレーティング・システムおよびデバイスのマネージの論理ボリューム・ストレージのセクションで、論理ボリューム・マネージャー、物理ボリューム、論理ボリューム、ボリューム・グループ、編成、データ保全性の確認、および割り当て特性について説明しています。

Web-based System Manager のインストールについては、 *AIX 5L Version 5.3 Web-based System Manager 管理ガイド* の第 2 章 インストールおよびシステム要件 のセクションを参照してください。

オペレーティング・システムおよびデバイスのマネージの System Management Interface Tool のセクションで、構造、メインメニュー、および SMIT で行なわれるタスクについて説明されています。

---

## rmm コマンド

### 目的

アクティブ状況からメッセージを除去します。

### 構文

```
rmm [ + Folder ] [ Messages ]
```

### 説明

**rmm** コマンドは、メッセージの名前を変更することによりアクティブ状況からそのメッセージを除去します。メッセージの名前を変更するために、システムは現行メッセージ番号の先頭に、(コンマ) を付けます。メッセージ・ハンドラー (MH) ・パッケージは非アクティブ・ファイルを使えません。ただし、システム・コマンドは引き続き非アクティブ・ファイルを操作できます。

**注:** **rmm** コマンドは、現行メッセージを変更しません。

非アクティブ・メッセージは定期的に削除する必要があります。**crontab** ファイル内にエントリーを入れれば、コンマで始まるファイルのすべてを自動的に削除できます。

### フラグ

**+Folder**           名前を変更するメッセージが入っているフォルダーを指定します。

*Messages* 名前を変更するメッセージを指定します。複数のメッセージ、メッセージの範囲、または単一メッセージを指定できます。下記の参照を使用します。

*Number* メッセージの数。

*Sequence*

ユーザーが指定するメッセージのグループ。指定できる値は、以下のとおりです。

**all** フォルダー内のすべてのメッセージ。

**cur** または **.** (ドット)

現行メッセージ。これはデフォルトです。

**first** フォルダー内の最初のメッセージ。

**last** フォルダー内の最後のメッセージ。

**next** 現行メッセージの次のメッセージ。

**prev** 現行メッセージの前のメッセージ。

**-help** コマンド構文、使用可能なスイッチ (トグル)、およびバージョン情報をリストします。  
注: MH の場合は、このフラグ名を省略することはできません。

## 例

1. 現行フォルダー内の現行メッセージを除去するには、以下のように入力します。

```
rmm
```

2. メッセージ 2 から 5 までを sales フォルダーから除去するには、以下のように入力します。

```
rmm +sales 2-5
```

## プロファイル・エントリー

以下のエントリーは、*UserMhDirectory*.**mh\_profile** に入力します。

**Current-Folder:** デフォルトの現行フォルダーを設定します。  
**Path:** *UserMhDirectory* を指定します。  
**rmmproc:** メッセージをフォルダーから除去するときに使用するプログラムを指定します。

## ファイル

**\$HOME/mh\_profile** MH ユーザー・プロファイルが入っています。  
**/usr/bin/rmm** **rmm** コマンドが入っています。

## 関連情報

**crontab** コマンド、**rmf** コマンド。

**.mh\_alias** ファイル・フォーマット、**.mh\_profile** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ のメール・アプリケーションのセクション。

---

## rmnamsv コマンド

### 目的

ホスト上で TCP/IP に基づくネーム・サービスの構成解除を行います。

### 構文

**rmnamsv** [ **-f** | **-F** *FileName* ]

### 説明

**rmnamsv** 高水準コマンドは、ホストで TCP/IP ベースのネーム・サービスを構成解除します。クライアントとして機能するホストのネーム・サービスを構成解除できます。

クライアント用のネーム・サービスを構成解除するために、**rmnamsv** コマンドは **namerslv** 低水準コマンドをコールして、**/etc/resolv.conf** ファイル内のエントリを構成解除するか、**/etc/resolv.conf** ファイルをデフォルト・ファイル名またはユーザー指定のファイル名に名前変更します。

Web-based System Manager のネットワーク・アプリケーションを使用して、ネットワーク特性を変更することができます。

また、System Management Interface Tool (SMIT) **smit rmnamerslv** 高速パスを使用してこのコマンドを実行することもできます。

### フラグ

|                           |                                                                                                 |
|---------------------------|-------------------------------------------------------------------------------------------------|
| <b>-F</b> <i>FileName</i> | システム構成データベースを <i>FileName</i> で指定されたファイル名に名前変更します。                                              |
| <b>-f</b>                 | <b>/etc/resolv.conf</b> ファイル名を名前変更するときに、デフォルト・ファイル名 ( <b>/etc/resolv.conf.sv</b> ) を使うことを指定します。 |

### ファイル

**/etc/resolv.conf** デフォルトのシステム構成データベースが入っています。

### 関連情報

ネットワークおよびコミュニケーションのマネージ の命名のセクション。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

**namerslv** コマンド。

---

## rmnfs コマンド

### 目的

システムの構成を変更して、実行中の NFS デーモンを停止させます。

## 構文

`/usr/sbin/rmnfs [ -I | -N | -B ]`

## 説明

**rmnfs** コマンドは、システムの再始動時に **/etc/rc.nfs** ファイルが実行されないように、システムの現行構成を変更します。また、コマンドに現在実行中の NFS デーモンを停止するように指示できます。

## フラグ

- B inittab** ファイル内のエントリを除去して、現在実行中の NFS デーモンを停止します。このフラグがデフォルトです。
- I inittab** 内の、システム再始動時に NFS デーモンを始動するエントリを除去します。
- N inittab** ファイルを変更せずに即座に NFS デーモンを停止させます。

## 例

すべての NFS デーモンを即座に停止させるには、以下のように入力します。

```
rmnfs -N
```

このコマンドは **inittab** ファイルを変更しません。

## 関連情報

**chnfs** コマンド、 **mknfs** コマンド、 **rmnfsexp** コマンド、 **rmnfsmnt** コマンド。

ネットワークおよびコミュニケーションのマネージ の NFS コマンドのリストのセクション。

ネットワークおよびコミュニケーションのマネージ のネットワーク・ファイルシステム (NFS) の概要のセクション。

---

## rmnfsexp コマンド

### 目的

NFS クライアントのディレクトリーをアンエクスポートします。

### 構文

`/usr/sbin/rmnfsexp -d Directory [ -V Exported Version ] [ -f Exports_file ] [ -I | -B | -N ] [ -F ]`

### 説明

**rmnfsexp** は、NFS クライアントのエクスポート・リストからエントリを除去します。このコマンドは、**exportfs** コマンドを始動して、指定されたディレクトリーをアンエクスポートします。**/etc/exports** ファイルにエントリが存在する場合は、そのエントリが除去されます。

### フラグ

- d Directory** アンエクスポートしたいディレクトリーを指定します。

|                                   |                                                                                                  |
|-----------------------------------|--------------------------------------------------------------------------------------------------|
| <b>-f</b> <i>Exports_File</i>     | <b>/etc/exports</b> ファイル以外のエクスポート・ファイルを使用する場合は、そのファイルの絶対パス名を指定します。                               |
| <b>-l</b>                         | コマンドに対して、 <b>exportfs</b> コマンドを実行せずに <b>/etc/exports</b> ファイルからエントリーを除去するように指示します。               |
| <b>-B</b>                         | 指定されたディレクトリーのエントリーを <b>/etc/exports</b> ファイルで除去し、 <b>exportfs</b> コマンドを実行してエクスポートを除去します。         |
| <b>-N</b>                         | <b>exportfs</b> コマンドを呼び出して、即座にディレクトリーをアンエクスポートします。このフラグを使用すると、 <b>/etc/exports</b> ファイルは修正されません。 |
| <b>-V</b> <i>Exported Version</i> | ディレクトリーのアンエクスポートに使用するバージョンを指定します。有効なバージョン番号は 2、3、および 4 です。                                       |
| <b>-F</b>                         | ディレクトリーのアンエクスポートを強制的に行います。                                                                       |

## 例

1. 即座にディレクトリーをアンエクスポートするには、以下のコマンドを入力します。

```
rmnfsexp -d /usr -N
```

この例では、**/usr** ディレクトリーが即座にアンエクスポートされます。

2. 即座に、また、システムを再始動するたびにディレクトリーをアンエクスポートするには、以下のコマンドを入力します。

```
rmnfsexp -d /home/guest -B
```

3. **/etc/exports** ファイル以外のエクスポート・ファイルから即座にディレクトリーをアンエクスポートするには、以下のコマンドを入力します。

```
rmnfsexp -d /usr -f /etc/exports.other -N
```

4. バージョン 3 としてエクスポートされた **/common/documents** ディレクトリーをアンエクスポートするには、次のコマンドを入力します。

```
rmnfsexp -d /common/documents -V 3
```

## ファイル

|                      |                                  |
|----------------------|----------------------------------|
| <b>/etc/exports</b>  | サーバーがエクスポートできるディレクトリーをリストします。    |
| <b>/etc/xtab</b>     | 現在エクスポートされているディレクトリーのリスト。        |
| <b>/etc/netgroup</b> | ネットワーク内の各ユーザー・グループに関する情報が入っています。 |

## 関連情報

**chnfsexp** コマンド、 **exportfs** コマンド、 **mknfsexp** コマンド。

ネットワークおよびコミュニケーションのマネージ の NFS コマンドのリストのセクション。

ネットワークおよびコミュニケーションのマネージ のネットワーク・ファイルシステム (NFS) の概要のセクション。

---

## rmnfsmnt コマンド

### 目的

NFS のマウントを除去します。



## 構文

```
/usr/sbin/rmnfsmnt -f PathName [ -l | -B | -N ]
```

## 説明

**rmnfsmnt** コマンドは、**/etc/filesystems** ファイルから該当するエントリーを除去して、指定されたファイルシステムのマウントを解除します。**-N** フラグを付けると、**rmnfsmnt** コマンドは **/etc/filesystems** ファイルを修正せずにファイルシステムのマウントを解除します。

## フラグ

- B** **/etc/filesystems** ファイル内のエントリーを除去して、ディレクトリーのマウントを解除します。**/etc/filesystems** ファイルにエントリーが存在しなければ、フラグはファイルに変更を加えません。ファイルシステムが現在マウントされていない場合は、フラグはマウントを解除しようとしません。このフラグがデフォルトです。
- f PathName** NFS でマウントされたファイルシステムのパス名を指定します。
- l** パス名で指定されたエントリーを **/etc/filesystems** ファイルから除去します。
- N** **/etc/filesystems** ファイルを修正せずに、指定されたディレクトリーのマウントを解除します。

## 例

1. ファイルシステムのマウントを解除するには、以下のように入力します。

```
rmnfsmnt -f /usr/man -N
```

この例では、**/usr/man** ファイルシステムがマウントを解除されます。

2. ファイルのマウントを除去するには、以下のように入力します。

```
rmnfsmnt -f /usr/local/man -B
```

この例では、**/usr/local/man** ファイルのマウントが除去されます。

## ファイル

**/etc/filesystems** システムの再始動時にマウントするリモート・ファイルシステムをリストします。

## 関連情報

**chnfsmnt** コマンド、**mknfsmnt** コマンド、**mount** コマンド、**umount** コマンド。

ネットワークおよびコミュニケーションのマネージ のネットワーク・ファイルシステム (NFS) の概要のセクション。

ネットワークおよびコミュニケーションのマネージ の NFS コマンドのリストのセクション。

---

## rmnfsproxy コマンド

### 目的

プロキシが使用可能化されている Caches の構成済み、マウント済みのインスタンスを除去します。

## 構文

`/usr/sbin/rmnfsproxy` *Cachefs\_mount\_point*

## 説明

指定された Cachefs マウントがアンマウントされます。 対応する NFS クライアント・マウントもアンマウントされます。最終的に、ローカル・ファイル・システムに作成されたすべてのキャッシュ情報が除去されます。

注: その Cachefs インスタンスが NFS エクスポートのものである場合は、**rmnfsproxy** の実行の前に、そのインスタンスをアンエクスポートする必要があります。

## パラメーター

*Cachefs\_mount\_point*

除去対象の、プロキシーが使用可能化された Cachefs インスタンスがマウントされた場所を指定します。

## 終了状況

0  
>0

コマンドは正常に実行されました。  
エラーが発生しました。

## 例

1. 既に構成済みの `/proj1_cached` Cachefs インスタンスを除去するには、次のように入力します。

```
rmnfsproxy /proj1_cached
```

## 位置

`/usr/sbin/rmnfsproxy`

## 関連情報

**mount** コマンド、**mknfsproxy** コマンド。

---

## rmnotify コマンド

### 目的

通知オブジェクト・クラスの通知メソッド定義を除去します。

### 構文

**rmnotify** -n *NotifyName*

### 説明

**rmnotify** コマンドは、通知オブジェクト・クラスの通知メソッド定義を除去します。

## フラグ

**-n** *NotifyName* 除去する通知メソッド定義を指定します。*NotifyName* 名が通知オブジェクト・クラス内に存在していなければ、**rmnotify** コマンドは異常終了します。

## ファイル

**/etc/objrepos/SRCnotify** SRC 通知メソッド・オブジェクト・クラスを指定します。

## 関連情報

**lssrc** コマンド、**mknotify** コマンド、**mkssys** コマンド、**rmssys** コマンド。

オペレーティング・システムおよびデバイスのマネージのシステム・リソース・コントローラーのセクション。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」のプログラマーのためのシステム・リソース・コントローラー (SRC) の概要のセクション。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」の SRC オブジェクトについてのセクション。

---

## rmpath コマンド

### 目的

システムから MPIO 可能デバイスへのパスを除去します。

### 構文

```
rmpath [ -l Name ] [ -p Parent ] [ -w Connection ] [ -d ]
```

```
rmpath -h
```

### 説明

**rmpath** コマンドは、指定されたターゲット・デバイス (**-l** *Name*) に関連付けられた、1 つまたは複数のパスを構成解除し、さらに定義解除を行う可能性もあります。除去されるパスのセットは、**-l** *Name*、**-p** *Parent* および **-w** *Connection* フラグの組み合わせにより決まります。このコマンドを実行した結果、デバイスに関連付けられたすべてのパスが構成解除または定義解除される場合は、このコマンドはエラーで終了し、パスの構成解除または定義解除は行われません。このような場合は、ターゲット・デバイス自体を構成解除または定義解除するのではなく、**rmdev** コマンドを使用する必要があります。

デフォルトのアクションは、指定されたそれぞれのパスを構成解除しますが、システムからそれを完全に除去することはしません。**-d** フラグを指定すると、**rmpath** コマンドは、システムからパス定義を (必要ならば) 構成解除し、さらに除去または削除します。

**rmpath** コマンドは終了時に、状況メッセージを表示します。パスを構成解除する時に、このコマンドは、いくつかのパスを構成解除できたけれども構成解除できないパスがあった (例えば、入出力処理を実行中のパスは構成解除できない) ということが起こります。

**rmpath** コマンドは、操作の結果についての状況メッセージを提供します。次のいずれかのフォーマットのメッセージが生成されます。

#### **path [ defined | deleted ]**

このメッセージは、1つのパスが正常に構成解除または定義解除された場合に表示されます。パスが正常に構成されている場合、メッセージ「path available」が表示されます。パスが正常に構成されていない場合で、かつメソッドから明示的なエラー・コードが戻されない場合、メッセージ「path defined」が表示されます。

#### **paths [defined | deleted]**

このメッセージは、複数のパスが指定され、すべてのパスが正常に構成解除または定義解除された場合に表示されます。 **-d** フラグを指定しない場合、メッセージは「paths defined」になります。 **-d** フラグを指定すると、メッセージは「paths deleted」になります。

#### **some paths [ defined | deleted ]**

このメッセージは、複数のパスを指定したけれども、そのいくつかだけが正常に構成解除または定義解除された場合に表示されます。 **-d** フラグを指定しない場合、メッセージは「some paths defined」になります。 **-d** フラグを指定すると、メッセージは「some paths deleted」になります。

#### **no paths processed**

このメッセージは、選択基準を満たすパスが見つからなかった場合に生成されます。

## フラグ

|                      |                                                                                                                           |
|----------------------|---------------------------------------------------------------------------------------------------------------------------|
| <b>-d</b>            | 指定されたパスをシステムから削除することを指示します。                                                                                               |
| <b>-h</b>            | コマンド使用方法メッセージを表示します。                                                                                                      |
| <b>-l Name</b>       | パスを除去するターゲット・デバイスの論理デバイス名を指定します。除去するパスは、 <b>-p</b> および <b>-w</b> フラグを使用して修飾されます。                                          |
| <b>-p Parent</b>     | 除去するパスを修飾するときに使用する親デバイスの論理デバイス名を示します。デバイスへのすべてのパスをこのコマンドで除去することはできませんから、このフラグまたは <b>-w</b> フラグ、あるいはその両方のフラグを指定しなければなりません。 |
| <b>-w Connection</b> | 除去するパスを修飾するために使用する接続情報を示します。デバイスへのすべてのパスをこのコマンドで除去することはできないので、このフラグまたは <b>-p</b> フラグ、またはその両方のフラグを指定する必要があります。             |

## セキュリティ

特権制御: root ユーザーおよびシステム・グループのメンバーのみが、このコマンドの実行権を持ちます。

監査イベントは次のとおりです。

| イベント       | 情報                                      |
|------------|-----------------------------------------|
| DEV_Change | rmpath,Unconfigure,<unconfigure メソッド引数> |
| DEV_Change | rmpath,Undefine,<undefine メソッド引数>       |

## 例

1. 接続 **5,0** で、**scsi0** から **hdisk1** へのパスを構成解除するには、次のように入力します。

```
rmpath -l hdisk1 -p scsi0 -w "5,0"
```

次のようなメッセージが生成されます。

```
path defined
```

2. **scsi0** から **hdisk1** へのすべてのパスを構成解除するには、次のように入力します。

```
rmpath -l hdisk1 -p scsi0
```

すべてのパスが正常に構成解除されると、次のようなメッセージが生成されます。

```
paths defined
```

しかし、パスの一部だけが正常に構成解除された場合は、次のようなメッセージが生成されます。

```
some paths defined
```

3. 接続 **5,0** で、**scsi0** と **hdisk1** の間のパス定義を定義解除するには、次のように入力します。

```
rmpath -d -l hdisk1 -p scsi0 -w "5,0"
```

次のようなメッセージが生成されます。

```
path deleted
```

4. **scsi0** から **hdisk1** へのすべてのパスを構成解除するには、次のように入力します。

```
rmpath -d -l hdisk1 -p scsi0
```

次のようなメッセージが生成されます。

```
paths deleted
```

## ファイル

`/usr/sbin/rmpath`

`rmpath` コマンドが含まれています。

## 関連情報

`chpath` コマンド、`lspath` コマンド、`mkpath` コマンド。

---

## rmprtsv コマンド

### 目的

クライアント・マシンまたはサーバー・マシンで印刷サービスの構成を解除します。

### 構文

```
rmprtsv { -c | -s } [ -T | -U | -A ] [ -h "HostName ..." | -H FileName ] [ -q "QEntry ..." ] [ -q QEntry -v "DeviceName ..." ]
```

### 説明

`rmprtsv` 高水準コマンドは、クライアント・マシンまたはサーバー・マシンで印刷サービスの構成解除を行います。

クライアント用の印刷サービスの構成解除を行うために、**rmprtsv** コマンドは、**rmque** および **rmquedev** コマンドをコールし、クライアントのプール・キューを使用不可にして、**/etc/qconfig** ファイル内の該当するエントリーを除去します。

サーバー用の印刷サービスの構成解除を行うために、**rmprtsv** コマンドは以下のプロシーチャーを実行します。

1. **stopsrc** コマンドをコールして、**lpd** および **qdaemon** サーバーの活動を停止します。
2. **ruser** 低水準コマンドをコールして、印刷サーバー上のリモート・ユーザーの構成解除を行います。
3. **rmque** および **rmquedev** コマンドをコールして、プーラーとそのデバイス・キューの構成解除を行って、サーバーの **/usr/lib/lpd/qconfig** ファイル内の該当するエントリーを削除します。

## フラグ

|                                  |                                                                                                      |
|----------------------------------|------------------------------------------------------------------------------------------------------|
| <b>-A</b>                        | 指定されたエントリーを <b>/etc/qconfig</b> ファイルから除去しますが、印刷サービスを完全に構成解除するわけではありません。                              |
| <b>-c</b>                        | クライアント・マシン用の印刷サービスの構成解除を行います。 <b>-q</b> フラグを <b>-c</b> フラグとともに使ってください。                               |
| <b>-H</b> <i>FileName</i>        | 印刷サービス用にそのまま構成しておくホスト名のリストが入っているファイル名を指定します。                                                         |
| <b>-h</b> <i>"HostName..."</i>   | 印刷サービスを使えないリモート・ホスト名のリストを指定します。キュー・システムでは、マルチバイトのホスト名はサポートされていないので注意が必要です。                           |
| <b>-q</b> <i>"QEntry..."</i>     | <b>/etc/qconfig</b> ファイルから除去したいエントリーのリストを指定します。                                                      |
| <b>-s</b>                        | サーバー・マシン用の印刷サービスの構成解除を行います。 <b>-s</b> フラグとともに、 <b>-h</b> 、 <b>-H</b> 、および <b>-q</b> フラグを使わなければなりません。 |
| <b>-T</b>                        | 印刷サービスを停止しますが、印刷サービスの完全な構成解除を行うわけではありません。                                                            |
| <b>-U</b>                        | 指定されたりモート・ユーザーを印刷サーバーで除去しますが、印刷サービスの完全な構成解除を行うわけではありません。                                             |
| <b>-v</b> <i>"DeviceName..."</i> | <b>qconfig</b> ファイル上のデバイス・スタンザ名のリストを指定します。 <b>-q</b> <i>QEntry</i> フラグとともに使用してください。                  |

## ファイル

**/etc/qconfig**            プリンター・キュー・システムの構成情報が入っています。

## 関連情報

**rmque** コマンド、 **rmquedev** コマンド、 **ruser** コマンド。

**lpd** デーモン、 **qdaemon** デーモン。

---

## rmpps コマンド

### 目的

アクティブでないページング・スペースを除去します。

## 構文

**rmpps** [ -t *ps\_helper*] *PagingSpace*

## 説明

**rmpps** コマンドは、アクティブでないページング・スペースを除去します。 *PagingSpace* パラメーターは、除去するページング・スペース名を指定します。この名前は、実際にはページング・スペースが存在する論理ボリューム名です。

NFS ページング・スペースの場合、*PagingSpace* パラメーターには除去するページング・スペース名を指定します。デバイスと、このページング・スペースに対応するデバイス定義がシステムから除去されます。ページングに使われるファイルが実際に存在する NFS サーバーでは何も変更されません。

**-t** フラグを指定すると、その引数はサード・パーティーのヘルパー実行可能モジュールであると仮定されます。ヘルパー実行可能モジュールが `/sbin/helpers/pagespace` パスに存在する場合、**rmpps** コマンドを指定するために **-r** フラグを受け渡して `spawn` されます。ヘルパー実行可能モジュールがゼロを戻した場合、`/etc/swapspace` ディレクトリーはそれによって変更されます。実行可能モジュールはページング・スペースの除去に注意する必要があります。名前が指定されたヘルパーが `/sbin/helpers/pagespace` ディレクトリーに存在しない場合、**rmpps** コマンドは使用方法エラーを表示します。ヘルパー実行可能モジュールは、正常完了した場合は 0、失敗に終わった場合は非ゼロを指定して終了する必要があります。

アクティブ・ページは、最初に **swapoff** コマンドを使用して非活動化することにより、除去することができます。

Web-based System Manager (wsm) のファイルシステム・アプリケーションを使用してファイルシステム特性を変更できます。

## フラグ

**-t** `/sbin/helpers/pagespace` ディレクトリーの下にあるヘルパー・プログラムを使用することを指定します。

*ps\_helper*

サード・パーティー・デバイスに対するヘルパー・プログラムの名前。

## 例

1. PS01 ページング・スペースを除去するには、次のコマンドを実行してください。

```
rmpps PS01
```

これにより、PS01 ページング・スペースが除去されます。

2. ヘルパー・プログラム `foo` を使用して PS01 ページング・スペースを除去するには、次のコマンドを実行してください。

```
rmpps -t foo PS01
```

これにより、PS01 ページング・スペースが除去されます。

## ファイル

`/etc/swapspace`

**swapon -a** コマンドによってアクティブにされるページング・スペース・デバイスを指定します。



## 関連情報

**lsps** コマンド、**mkps** コマンド、**chps** コマンド、**rmlv** コマンド、**swap** コマンド、**swapon** コマンド、**swapoff** コマンド。

オペレーティング・システムおよびデバイスのマネージのページング・スペースのセクションで、ページング・スペースとページング・スペースの割り振りポリシーについて説明しています。

オペレーティング・システムおよびデバイスのマネージのファイルシステムのセクションで、ファイルシステムのタイプ、管理、構造、および保守について説明しています。

オペレーティング・システムおよびデバイスのマネージの論理ボリューム・ストレージのセクションで、論理ボリューム・マネージャー、物理ボリューム、論理ボリューム、ボリューム・グループ、編成、データ保全性の確認、および割り当て特性の理解について説明しています。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件 のセクションを参照してください。

オペレーティング・システムおよびデバイスのマネージのSystem Management Interface Toolのセクションで、構造、メインメニュー、および SMIT で行なわれるタスクについて説明されています。

---

## rmqos コマンド

### 目的

システムの構成を変更して、QoS サポートを除去します。

### 構文

```
/usr/sbin/rmqos [ -I | -N | -B ]
```

### 説明

**rmqos** コマンドは、システムの現行構成を変更して、サービス品質 (QoS) サポートを除去します。

### フラグ

- B** システム始動時に QoS を使用可能にする **inittab** ファイルのエントリーを除去して、QoS デーモンを停止します。このフラグがデフォルトです。
- I** システム始動時に QoS を使用可能にする **inittab** ファイルのエントリーを除去しますが、現在実行している QoS サブシステムには影響を与えません。
- N** **inittab** ファイルを変更せずに、QoS サポートをすぐに使用不可にします。

### ファイル

**inittab** システムの初期化プロセスを制御します。  
**/etc/rc.qos** QoS デーモンの始動スクリプトが入っています。

## 関連情報

**mkqos** コマンド。

ネットワークおよびコミュニケーションのマネージ の **TCP/IP** のサービス品質 (**QoS**) のセクション。

---

## rmque コマンド

### 目的

システムからプリンター・キューを除去します。

### 構文

**rmque** **-q** *Name*

### 説明

**rmque** コマンドは、**/etc/qconfig** ファイル内の **-q** フラグで指定されたキュー・スタanzasを削除して、システム構成からキューを除去します。 このコマンドを入力する前に、**rmquedev** コマンドを使ってすべてのキュー・デバイスを削除しなければなりません。

Web-based System Manager (wsm) のプリンター・キュー・アプリケーションを使用してプリンター特性を変更できます。

また、System Management Interface Tool (SMIT) **smit rmque** 高速パスを使ってこのコマンドを実行することもできます。

**推奨:** **/etc/qconfig** ファイルを編集するには、**chque**、**mkque**、**rmque**、**chquedev**、**mkquedev**、および **rmquedev** の各コマンドまたは SMIT を使用します。さらに、これらのコマンドを実行する時間帯は、ピークが弱まった時、またはオフピーク時にすることをお勧めします。

手動操作による **/etc/qconfig** ファイルの編集が必要な場合、まず **enq -G** コマンドを発行し、すべてのジョブがプロセスされた後、キューイング・システムと **qdaemon** を一時停止状態にしてください。次に、**/etc/qconfig** ファイルを編集し、新しい構成を使用して **qdaemon** を再始動できます。

### フラグ

**-q** *Name*      除去するキューの名前を指定します。

### 例

プリンター・キュー **lp0** を除去するには、以下のように入力します。

```
rmque -q lp0
```

### ファイル

**/usr/bin/rmque**  
**/etc/qconfig**

**rmque** コマンドが入っています。  
構成ファイルが入っています。

## 関連情報

**chque** コマンド、 **lsque** コマンド、 **mkque** コマンド、 **rmqueuedev** コマンド、 **smit** コマンド。

**/etc/qconfig** ファイル。

プリンターおよび印刷 の 印刷の管理 のセクション。

プリンターおよび印刷 の 印刷スプーラー のセクション。

プリンターおよび印刷 の 印刷キューの削除 のセクション。

Web-based System Manager のインストールについては、 *AIX 5L Version 5.3 Web-based System Manager 管理ガイド* の第 2 章 インストールおよびシステム要件のセクションを参照してください。

---

## rmqueuedev コマンド

### 目的

システムからプリンター・キュー・デバイスまたはプロッター・キュー・デバイスを除去します。

### 構文

```
rmqueuedev -d Name -q Name
```

### 説明

**rmqueuedev** コマンドは、 **-d** フラグで指定されたデバイス・スタンザを **/etc/qconfig** ファイルから削除して、システム構成からプリンター・キュー・デバイスまたはプロッターのキュー・デバイスを除去します。また、デバイス **Name** のエントリーを削除して、キュー・スタンザの **Device=DeviceName1,DeviceName2,DeviceName3** の行を修正します。

Web-based System Manager (wsm) のプリンター・キュー・アプリケーションを使用してプリンター特性を変更できます。

また、System Management Interface Tool (SMIT) **smit rmqueuedev** 高速パスを使って実行することもできます。

**推奨:** **/etc/qconfig** ファイルを編集するには、**chque**、**mkque**、**rmque**、**chqueuedev**、**mkqueuedev**、および **rmqueuedev** コマンドまたは SMIT を使用します。さらに、これらのコマンドを実行する時間帯は、ピークが弱まった時、またはオフピーク時にすることをお勧めします。

手動操作による **/etc/qconfig** ファイルの編集が必要な場合、まず **enq -G** コマンドを発行し、すべてのジョブがプロセスされた後、キューイング・システムと **qdaemon** を一時停止状態にしてください。次に、**/etc/qconfig** ファイルを編集し、新しい構成を使用して **qdaemon** を再始動できます。

### フラグ

**-d Name**      **qconfig** ファイルから削除するデバイス・スタンザの **Name** を指定します。  
**-q Name**      前のキュー・スタンザで修正するデバイスの **Name** を指定します。

## 例

**/etc/qconfig** ファイルから `loc` を削除して、前のキュー・スタンザ `lpq` 内で "DEVICE =" スタンザを修正するには、以下のように入力します。

```
rmqueuedev -q lpq -d loc
```

## ファイル

**/usr/bin/rmqueuedev** **rmqueuedev** コマンドが入っています。  
**/etc/qconfig** 構成ファイル。

## 関連情報

**chqueuedev** コマンド、**lsqueuedev** コマンド、**mkqueuedev** コマンド、**rmque** コマンド、**smit** コマンド。

**/etc/qconfig** ファイル。

プリンターおよび印刷 の 印刷の管理 のセクション。

プリンターおよび印刷 の 印刷スプーラー のセクション。

プリンターおよび印刷 の 印刷キューの削除 のセクション。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

---

## rmramdisk コマンド

### 目的

**mkramdisk** コマンドが作成した RAM ディスクを除去します。

### 構文

```
rmramdisk ram_disk_name
```

### 説明

**rmramdisk** コマンドは、指定された RAM ディスク、およびその RAM ディスク用に作成されたデバイス・スペシャル・ファイルを除去します。システムが再起動されると、RAM ディスクも除去されます。デバイス・スペシャル・ファイルは、**rmramdisk** コマンドを使用してのみ、除去することができます。

### パラメーター

*ram\_disk\_name* メモリーから除去される特定の RAM ディスクの名前。指定されていない場合は、エラーが戻されます。RAM ディスクの名前は **rramdiskx** の形式で表されます。ここで、*x* は論理 RAM ディスク番号 (0 から 63) です。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0 正常終了。
- >0 エラーが発生しました。

## 例

```
# ls -l /dev/*ramdisk2
brw----- 1 root    system    38, 0 Aug 01 05:52 /dev/ramdisk2
crw----- 1 root    system    38, 0 Aug 01 05:52 /dev/rramdisk2
```

ramdisk2 を除去するには、次のように入力します。

```
# rmdir ramdisk2

# ls -l /dev/*ramdisk2
ls: 0653-341 The file /dev/*ramdisk2 does not exist.
```

## ファイル

`/usr/sbin/rmdir` `rmdir` コマンドが入っています。

## 関連情報

`mkramdisk` コマンド、および `ls` コマンド。

---

## rmresponse コマンド

### 目的

応答を除去します。

### 構文

```
rmresponse [-f] [-q] [-h] [-TV] response[:node_name]
```

### 説明

**rmresponse** コマンドは *response* パラメーターで指定された応答を除去します。除去する応答は、既に存在するものでなければなりません。条件とリンクされていても応答を除去する必要がある場合は、**-f** フラグを指定します。これにより、応答、および条件とのリンクが強制的に除去されます。**-f** フラグを指定しない場合は、条件とのリンクが存在しても、応答は除去されません。このコマンドは条件を除去しません。

システム・ソフトウェアを適切に機能させるために特定の応答が必要な場合、その応答はロックされることがあります。ロックされた応答はアンロックされない限り変更したり、取り除くことはできません。

**rmresponse** コマンドで指定した応答がロックされた場合、その応答は除去されません。代わりに、応答がロックされたという通知のエラーが作成されます。応答をアンロックするには、**chresponse** コマンドの **-U** フラグを使用することができます。ただし、応答はシステム・ソフトウェアが適切に機能するために必須であり、通常ロックされているので、これをアンロックする際は注意が必要です。

## フラグ

- f 応答が条件とリンクされていても、応答を強制的に除去します。条件とのリンクも応答同様に除去されますが、条件は除去されません。
- q *response* が存在しない場合、エラーを戻しません。
- h コマンドの使用ステートメント (使用法) を標準出力に書き込みます。
- T コマンドのトレース・メッセージを標準エラーに書き込みます。これはソフトウェア・サービス担当者のみが使用します。
- V コマンドの詳細メッセージを標準出力に書き込みます。

## パラメーター

- response* 除去する、定義済みの応答の名前を指定します。
- node\_name* 応答が定義されているクラスター内のノードを指定します。*node\_name* を指定しないと、ローカル・ノードが使用されます。*node\_name* は、CT\_MANAGEMENT\_SCOPE 環境変数により定義された有効範囲内のノードです。

## セキュリティ

**rmresponse** を実行するユーザーは、**IBM.EventResponse** リソース・クラスに書き込み権限が必要です。権限は、相手のシステムのアクセス制御リスト (ACL) ファイルに指定します。

## 終了状況

- 0 コマンドが正常に実行されました。
- 1 RMC でエラーが起きました。
- 2 コマンド行インターフェース・スクリプトでエラーが起きました。
- 3 コマンド行に入力したフラグが誤りです。
- 4 コマンド行に入力したパラメーターが誤りです。
- 5 誤ったコマンド行入力により、エラーが起きました。

## 環境変数

### CT\_CONTACT

リソース・モニターおよび制御 (RMC) デーモンを使用したセッションを行うシステムを指定します。CT\_CONTACT をホスト名または IP アドレスに設定すると、コマンドは指定されたホストの RMC デーモンと連絡します。CT\_CONTACT を設定しないと、コマンドは、そのコマンドが実行されているローカル・システムの RMC デーモンと連絡します。RMC デーモン・セッションのターゲットと管理有効範囲により、処理されるリソース・クラスまたはリソースが決められます。

### CT\_IP\_AUTHENT

CT\_IP\_AUTHENT 環境変数が存在する場合、RMC デーモンは、IP ベース・ネットワーク認証を使用して、CT\_CONTACT 環境変数が設定されている IP アドレスで指定されたシステム上の RMC デーモンに連絡します。CT\_IP\_AUTHENT は、CT\_CONTACT が IP アドレスに設定されている場合にのみ意味を持ちます。ドメイン・ネーム・システム (DNS) サービスには依存しません。

## CT\_MANAGEMENT\_SCOPE

イベント応答リソース・マネージャー (ERRM) のリソースの処理で、RMC デーモンとのセッションに使用する管理有効範囲を指定します。管理有効範囲は、リソースを処理できる、可能なターゲット・ノードのセットを決定します。有効な値は以下のとおりです。

- 0 *local* 有効範囲を指定します。
- 1 *local* 有効範囲を指定します。
- 2 *peer domain* (ピア・ドメイン) 有効範囲を指定します。
- 3 *management domain* (管理ドメイン) 有効範囲を指定します。

この環境変数を設定しないと、*local* 有効範囲が使用されます。

## 標準出力

**-h** フラグが指定されている場合は、このコマンドの使用状況の説明文が標準出力に書き込まれます。詳細メッセージはすべて、標準出力に書き込まれます。

## 標準エラー

トレース・メッセージはすべて、標準エラーに書き込まれます。

## 例

以下の例は、スタンドアロン・システムに適用されます。

1. "Broadcast event on-shift" という名前の応答定義を除去するには、次のコマンドを実行します。

```
rmresponse "Broadcast event on-shift"
```

2. 応答が条件とリンクされていても、"Broadcast event on-shift" という名前の応答定義を除去するには、次のコマンドを実行します。

```
rmresponse -f "Broadcast event on-shift"
```

この例は、管理ドメインに適用されます。

1. この例では、現行ノードは管理サーバーです。管理ノード **nodeB** 上の、"Broadcast event on-shift" という名前の応答定義を除去するには、次のコマンドを実行します。

```
rmresponse "Broadcast event on-shift":nodeB
```

この例は、ピア・ドメインに適用されます。

1. 管理ノード **nodeA** に定義されている、"Broadcast event on-shift" という名前の応答定義を除去するには、ドメイン内の任意のノードから次のコマンドを実行します。

```
rmresponse "Broadcast event on-shift":nodeA
```

## 位置

**/usr/sbin/rsct/bin/rmresponse**

## 関連情報

コマンド: **chresponse**、**lscondresp**、**lsresponse**、**mkcondresp**、**mkresponse**

ファイル: **rmccli**



---

## rmrole コマンド

### 目的

ロールを除去します。このコマンドは、AIX 4.2.1 以降にだけ適用されます。

### 構文

**rmrole** *Name*

### 説明

**rmrole** コマンドは、**/etc/security/roles** ファイルから *Name* パラメーターで識別されたロールを除去します。ロール名は、既に存在してはなりません。

このコマンドは、Web-based System Manager ユーザー・アプリケーション、または System Management Interface Tool (SMIT) を使用して実行できます。

### セキュリティー

アクセスされるファイルは次のとおりです。

|           |                          |
|-----------|--------------------------|
| モード       | ファイル                     |
| <b>rw</b> | /etc/security/roles      |
| <b>r</b>  | /etc/security/user.roles |

監査イベントは次のとおりです。

|                    |     |
|--------------------|-----|
| イベント               | 情報  |
| <b>ROLE_Remove</b> | ロール |

### 例

1. ManageObjects ロールを除去するには、以下のように入力します。

```
rmrole ManageObjects
```

### ファイル

|                                 |                    |
|---------------------------------|--------------------|
| <b>/etc/security/roles</b>      | ロールの属性が入っています。     |
| <b>/etc/security/user.roles</b> | ユーザーのロール属性が入っています。 |

### 関連情報

**chrole** コマンド、**lsrole** コマンド、**mkrole** コマンド、**chuser** コマンド、**lsuser** コマンド、**mkuser** コマンド。

セキュリティーのネットワークの保護のセクション。

セキュリティーのユーザー、ロール、およびパスワードのセクション。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

---

## rmrpdomain コマンド

### 目的

定義済みのピア・ドメインを除去します。

### 構文

```
rmrpdomain [-f] [-q] [-h] [-TV] peer_domain
```

### 説明

**rmrpdomain** コマンドは、*peer\_domain* パラメーターで指定されたピア・ドメイン定義を除去します。除去するピア・ドメインは、既に定義済みでなければなりません。このコマンドは、ピア・ドメインに定義されているノードで実行する必要があります。**rmrpdomain** をピア・ドメインに対してオンラインになっているノード上で実行すると、ピア・ドメインに対して定義されているノードのうち、このノードから到達できるすべてのノード上のピア・ドメイン定義が除去されます。ピア・ドメインに定義されているが、到達できないノードについては、そのノードのローカル・ピア・ドメイン定義は除去されません。ピア・ドメインがオンラインではない、またはノードがピア・ドメインに対してオンラインになっていない場合に、ローカル・ピア・ドメイン定義を除去するには、そのノード上で **rmrpdomain** コマンドを実行し、**-f** フラグを指定します。

ピア・ドメイン定義を除去する最も効率的な方法は、ピア・ドメインを確実にオンラインにすることです。その後で、ピア・ドメインにオンラインになっているノードから **rmrpdomain** コマンドを実行します。

**rmrpdomain** コマンドを実行するノードからアクセスできないノードがある場合、それらの各ノード上で **-f** フラグを指定して **rmrpdomain** コマンドを実行します。ノード自体が操作可能でない場合は、これは後で実行することができます。

ピア・ドメインの除去をサブシステムが拒否することをオーバーライドするため、**-f** フラグも使用する必要があります。サブシステムは、例えばピア・ドメイン・リソースが使用中の場合、要求を拒否する可能性があります。このような場合に **-f** フラグを指定すると、ピア・ドメイン定義を除去しなければならないことをサブシステムに指示します。

**rmrpdomain** コマンドに構成クォーラムは不要です。したがって、このコマンドは、小サブクラスターに対して発行した場合も正常に実行されます。後で、大サブクラスターがアクティブになるかもしれません。その場合でも、ドメインは除去されます。

### フラグ

- f**     ピア・ドメインを強制的に除去します。以下の場合、ピア・ドメイン定義を除去するためにはこの強制フラグが必要です。
  - ノードがピア・ドメインにオンラインになっていない場合にローカル・ノードから除去する。
  - サブシステムが、例えばリソースが割り当てられている時に、要求を拒否する場合。
- q**     抑止モードを指定します。ピア・ドメインが存在しない場合、コマンドはエラーを戻しません。
- h**     コマンドの使用ステートメント (使用法) を標準出力に書き込みます。
- T**     コマンドのトレース・メッセージを標準エラーに書き込みます。これはソフトウェア・サービス担当者のみが使用します。
- V**     コマンドの詳細メッセージを標準出力に書き込みます。

## パラメーター

`peer_domain` 除去する、定義されたピア・ドメインの名前を指定します。

## セキュリティ

`rmrpdomain` コマンドのユーザーは、ピア・ドメインに定義された各ノードの **IBM.PeerDomain** リソース・クラスに書き込み権限が必要です。デフォルトでは、ピア・ドメイン内のどのノード上の **root** も、構成リソース・マネージャーを経由して、このリソース・クラスに読み取りおよび書き込み権限を持ちます。

## 終了状況

- 0 コマンドが正常に実行されました。
- 1 RMC でエラーが起きました。
- 2 コマンド行インターフェース・スクリプトでエラーが起きました。
- 3 コマンド行に入力したフラグが誤りです。
- 4 コマンド行に入力したパラメーターが誤りです。
- 5 誤ったコマンド行入力により、エラーが起きました。
- 6 ピア・ドメイン定義が存在しません。

## 環境変数

### CT\_CONTACT

リソース・モニターおよび制御 (RMC) デーモンを使用したセッションを行うシステムを指定します。CT\_CONTACT をホスト名または IP アドレスに設定すると、コマンドは指定されたホストの RMC デーモンと連絡します。CT\_CONTACT を設定しないと、コマンドは、そのコマンドが実行されているローカル・システムの RMC デーモンと連絡します。RMC デーモン・セッションのターゲットと管理有効範囲により、処理されるリソース・クラスまたはリソースが決めます。

### CT\_IP\_AUTHENT

CT\_IP\_AUTHENT 環境変数が存在する場合、RMC デーモンは、IP ベース・ネットワーク認証を使用して、CT\_CONTACT 環境変数が設定されている IP アドレスで指定されたシステム上の RMC デーモンに連絡します。CT\_IP\_AUTHENT は、CT\_CONTACT が IP アドレスに設定されている場合にのみ意味を持ちます。ドメイン・ネーム・システム (DNS) サービスには依存しません。

## 制約事項

このコマンドを実行するノードは、ピア・ドメインに定義されている必要があり、またピア・ドメインに定義されたすべてのノードに到達できるべきです。ノードに到達できない場合、ノードのローカル・ピア・ドメイン定義は除去されません。

## 実装上の固有な条件

このコマンドは、AIX 用の Reliable Scalable Cluster Technology (RSCT) ファイルセットの一部です。

## 標準入力

`-f "-"` または `-F "-"` フラグが指定されているときは、このコマンドで 1 つ以上のノード名が標準入力から読み取られます。

## 標準出力

**-h** フラグが指定されている場合は、このコマンドの使用状況の説明文が標準出力に書き込まれます。詳細メッセージはすべて、標準出力に書き込まれます。

## 標準エラー

トレース・メッセージはすべて、標準エラーに書き込まれます。

## 例

1. ピア・ドメイン定義 **ApplDomain** (ここでは **nodeA**、**nodeB**、および **nodeC** が定義されており、これらのノードは **ApplDomain** にオンラインになっており、すべてがお互いに到達できる) を除去するには、次のコマンドを **nodeA**、**nodeB**、または **nodeC** で実行します。

```
rmrpdomain ApplDomain
```

2. **nodeD** がピア・ドメイン **ApplDomain** にオンラインになっていない (例えばピア・ドメインがオフラインになっている、またはピア・ドメインが存在しない) 場合に、**nodeD** 上のこのローカル・ピア・ドメイン定義を除去するには、次のコマンドを **nodeD** で実行します。

```
rmrpdomain -f ApplDomain
```

3. ピア・ドメイン定義 **ApplDomain** (ここでは **nodeA**、**nodeB**、および **nodeC** が定義されており、これらのノードは **ApplDomain** にオンラインになっており、すべてがお互いに到達できる) を除去し、サブシステムが要求を拒否するのを防ぐには、次のコマンドを **nodeA**、**nodeB**、または **nodeC** で実行します。

```
rmrpdomain -f ApplDomain
```

## 位置

**/usr/sbin/rsct/bin/rmrpdomain**

## ファイル

**/etc/services** ファイルが変更されます。

## 関連情報

ブック: 「*RSCT: Administration Guide*」(ピア・ドメイン操作についての説明)

コマンド: **addrpnode**、**lsrpdomain**、**lsrpnode**、**mkrpdomain**、**preprnode**、**rmrpnode**、**startrpdomain**、**stoprpdomain**

情報ファイル: **rmccli** (RMC 関連のコマンドの一般情報)

ファイル: **/etc/services**

---

## rmrpnode コマンド

### 目的

ピア・ドメイン定義から 1 つ以上のノードを除去します。

## 構文

```
rmrpnnode [-f] [-q] [-h] [-TV] node_name1 [node_name2 ...]
```

```
rmrpnnode -F { file_name | "-" } [-f] [-q] [-h] [-TV]
```

## 説明

**rmrpnnode** コマンドは、コマンドを実行するオンライン・ピア・ドメインから、1 つ以上のノードを除去します。このコマンドは、ノードを除去するピア・ドメインにオンラインになっているノードで実行する必要があります。除去するノードは、ピア・ドメインに対してオフラインになっている必要があります。コマンドを実行するノードからアクセス可能である必要があります。ノードをオフラインにするには、**stoprpnnode** コマンドを使用します。

**-f** フラグを指定すると、指定されたノードがピア・ドメインから強制的に除去されます。**-f** フラグが無指定の場合は、ドメインから 1 つ以上のノードを除去するには、過半数のノードがオンラインである必要があります。

## フラグ

- f** 指定されたノードをピア・ドメインから強制的に除去します。
- q** 抑止モードを指定します。指定されたノードがピア・ドメインにない場合、コマンドはエラーを戻しません。
- F { *file\_name* | "-" }**  
*file\_name* からノード名のリストを読み取ります。ファイルの 1 行ごとに 1 つのノード名が走査されます。**#** 記号は、その行の残りの部分 (**#** が 1 桁目にある場合はその行の全体) がコメントであることを示します。  
**-F "-"** を使用して、**STDIN** を入力ファイルとして指定します。
- h** コマンドの使用ステートメント (使用法) を標準出力に書き込みます。
- T** コマンドのトレース・メッセージを標準エラーに書き込みます。これはソフトウェア・サービス担当者のみが使用します。
- v** コマンドの詳細メッセージを標準出力に書き込みます。

## パラメーター

*node\_name1* [*node\_name2* ...] ピア・ドメイン定義から除去するノードのピア・ドメイン・ノード名を指定します。**rmrpnnode** コマンドを使用して 1 つ以上のノードを除去することができます。このノード名を指定する場合は、このノード名が **addrpnnode** コマンドまたは **mkrpdomain** コマンドで指定したのとまったく同一形式にする必要があります。ピア・ドメイン・ノード名をリストするには、**lsrpnnode** コマンドを実行します。

## セキュリティ

**rmrpnnode** コマンドのユーザーは、ピア・ドメインから除去する各ノードの **IBM.PeerNode** リソース・クラスに書き込み権限が必要です。デフォルトでは、ピア・ドメイン内のどのノード上の **root** も、構成リソース・マネージャーを経由して、このリソース・クラスに読み取りおよび書き込み権限を持ちます。

## 終了状況

- 0 コマンドが正常に実行されました。
- 1 RMC でエラーが起きました。
- 2 コマンド行インターフェース・スクリプトでエラーが起きました。
- 3 コマンド行に入力したフラグが誤りです。
- 4 コマンド行に入力したパラメーターが誤りです。
- 5 誤ったコマンド行入力により、エラーが起きました。
- 6 ピア・ドメインにノードが存在しません。

## 環境変数

### CT\_CONTACT

リソース・モニターおよび制御 (RMC) デーモンを使用したセッションを行うシステムを指定します。CT\_CONTACT をホスト名または IP アドレスに設定すると、コマンドは指定されたホストの RMC デーモンと連絡します。CT\_CONTACT を設定しないと、コマンドは、そのコマンドが実行されているローカル・システムの RMC デーモンと連絡します。RMC デーモン・セッションのターゲットと管理有効範囲により、処理されるリソース・クラスまたはリソースが決められます。

### CT\_IP\_AUTHENT

CT\_IP\_AUTHENT 環境変数が存在する場合、RMC デーモンは、IP ベース・ネットワーク認証を使用して、CT\_CONTACT 環境変数が設定されている IP アドレスで指定されたシステム上の RMC デーモンに連絡します。CT\_IP\_AUTHENT は、CT\_CONTACT が IP アドレスに設定されている場合にのみ意味を持ちます。ドメイン・ネーム・システム (DNS) サービスには依存しません。

## 制約事項

このコマンドは、ノードを除去するピア・ドメインにオンラインになっているノードで実行する必要があります。除去するノードは、ピア・ドメインにオフラインになっていなければなりません。

## 実装上の固有な条件

このコマンドは、AIX 用の Reliable Scalable Cluster Technology (RSCT) ファイルセットの一部です。

## 標準入力

**-f** "-" フラグを指定すると、このコマンドは 1 つ以上のノード名を標準入力から読み取ります。

## 標準出力

**-h** フラグが指定されている場合は、このコマンドの使用状況の説明文が標準出力に書き込まれます。詳細メッセージはすべて、標準出力に書き込まれます。

## 標準エラー

トレース・メッセージはすべて、標準エラーに書き込まれます。

## 例

ノード **nodeB** および **nodeC** のピア・ドメイン定義を、ピア・ドメイン **AppIDomain** から削除します。この時、**nodeA** は **AppIDomain** に定義されており、このドメインにオンラインになっているノードであり、**nodeB** および **nodeC** には **nodeA** から到達できる場合、次のコマンドを **nodeA** から実行します。

```
rmrpnode nodeB nodeC
```

## 位置

```
/usr/sbin/rsct/bin/rmrpnode
```

## 関連情報

ブック: 「*RSCT: Administration Guide*」(ピア・ドメイン操作についての説明)

コマンド: **addrpnode**、**lsrpnode**、**preprpnode**、**startrpnode**、**stoprpnode**

情報ファイル: **rmccli** (RMC 関連のコマンドの一般情報)

---

## rmrset コマンド

### 目的

システム・レジストリーから **rset** を除去します。

### 構文

```
rmrset rsetname
```

### 説明

**rmrset** コマンドは、システム・レジストリーから **rset** または排他的 **rset** (**xrset**) を除去します。**rmrset** コマンドは、**xrset** の削除に使用される場合は、システム上の対応する CPU の状態を汎用モードに変更します。**xrset** の削除には **root** 権限が必要です。

### パラメーター

**rsetname** システム・レジストリーから除去する **rset** の名前。名前は、*namespace* および **/** (スラッシュ) で区切られた *rsname* からなります。*namespace* および *rsname* は両方とも、255 文字まで指定できます。**rset** 名の文字セット制限についての追加情報は、**rs\_registername()** サービスを参照してください。

### セキュリティ

ユーザーは **root** 権限を持つか、または **CAP\_NUMA\_ATTACH** 機能と指定された **rset** への書き込み権限が必要です。

## 例

1. システム・レジストリーから **test/cpus0to7** を除去するには、次のように入力します。

```
rmrset test/cpus0to7
```



## ファイル

`/usr/bin/rmrset`

`rmrset` コマンドが入っています。

## 関連情報

`attachrset`、`detachrset`、`execrset`、`lsrset`、および `mkrset` コマンド。

---

## rmrsrc コマンド

### 目的

定義済みリソースを除去します。

### 構文

1 つ以上のリソースを除去するには...

- コマンド行で次のように入力します。

```
rmrsrc -s "selection_string" [ -a | -N { node_file | "-" } ] [-h] [-TV] resource_class
```

```
rmrsrc -r "resource_handle" [-h] [-TV]
```

- 入力ファイルでは次のように事前定義されています。

```
rmrsrc -f resource_data_input_file -s "selection_string" [ -a | -N { node_file | "-" } ] [-h] [-TV]  
resource_class
```

```
rmrsrc -f resource_data_input_file -r "resource_handle" [-h] [-TV]
```

コマンド引数の名前とデータ・タイプを表示するには、次のように入力します。

```
rmrsrc -l [-h] resource_class
```

### 説明

**rmrsrc** コマンドは、指定されたリソース・インスタンス (複数も可) を除去 (定義を解除) します。

**rmrsrc** コマンドは、RMC (リソース・モニターおよび制御) サブシステムに要求を出して、特定のリソース・インスタンスの定義を解除します。リソースのリソース・マネージャーはそのリソースを除去します。

このコマンドの上記の最初のフォーマットは、リソース・クラス名パラメーターおよび、**-s** フラグを使用して指定された選択文字列を必要とします。指定された選択文字列に一致する指定されたリソース・クラス内の全リソース除去されます。選択文字列が除去するリソースを複数識別した場合は、選択文字列に一致するリソースごとに一度ずつこのコマンドを実行するのと同じことです。

このコマンドの 2 番目の形式を使用することで、特定のリソースにリンクした実際のリソース・ハンドルをパラメーターとして指定できます。コマンドのこの形式は、スクリプト内で使用する必要があります。

*selection\_string* に複数のノード名を指定する代わりに、**-N** *node\_file* フラグを使用して、ノード名がファイルにあることを指示できます。標準入力からノード名を読み取るには、**-N** "-" を使用します。

システムにクラスター・システム・マネージメント (CSM) がインストールされている場合は、ノード名の値として CSM 定義のノード・グループを使用して、複数のノードを参照できます。CSM ノード・グループの処理と CSM **nodegrp** コマンドの使用については、「CSM: Administration Guide」および「CSM: Command and Technical Reference」を参照してください。

## フラグ

**-a** このコマンドをクラスター内のすべてのノードに適用することを指定します。クラスターの有効範囲は、**CT\_MANAGEMENT\_SCOPE** 環境変数により決まります。この環境変数が設定されていない場合、コマンドにとって有効な有効範囲になるまで、最初に管理ドメイン有効範囲 (存在する場合) を選択し、次にピア・ドメイン有効範囲 (存在する場合) を選択し、次にローカル有効範囲を選択します。コマンドは、最初に見つかった有効な有効範囲に対して、1 回実行されます。例えば、管理ドメインとピア・ドメインの両方が存在し、**CT\_MANAGEMENT\_SCOPE** が設定されていない **rmrsrc -a** は、管理ドメインに適用されます。この場合、ピア・ドメインに適用するには **CT\_MANAGEMENT\_SCOPE** を **2** に設定します。

**-f resource\_data\_input\_file**  
リソースの引数情報が含まれているファイルの名前を指定します。

**-I** コマンド引数とデータ・タイプをリストします。一部のリソース・マネージャーは、除去要求に渡された追加の引数を受け入れます。定義済みのコマンド引数およびコマンド引数値のデータ・タイプをリストするには、このフラグを指定してください。

**-N { node\_file | "-" }**  
ファイルまたは標準入力からノード名を読み取ることを指定します。ノード名がファイルにあることを示すには、**-N node\_file** を指定します。

- **node\_file** では 1 行に 1 つのノード名を指定します。
- 1 桁目の番号記号 (**#**) はその行がコメントであることを示します。
- ノード名の左のブランク文字はすべて無視されます。
- ノード名の右のブランク文字はすべて無視されます。

標準入力からノード名を読み取るには、**-N "-"** を使用します。

**CT\_MANAGEMENT\_SCOPE** 環境変数は、クラスターの有効範囲を決定します。

**CT\_MANAGEMENT\_SCOPE** が設定されていない場合、コマンドにとって有効な範囲になるまで、最初に管理ドメイン有効範囲 (存在する場合) を選択し、次にピア・ドメイン有効範囲 (存在する場合) を選択し、次にローカル有効範囲を選択します。コマンドは、最初に見つかった有効範囲に対して、1 回実行されます。例えば、管理ドメインとピア・ドメインの両方が存在する場合に **CT\_MANAGEMENT\_SCOPE** が設定されていないと、このコマンドは管理ドメインに適用されません。このコマンドをピア・ドメインに適用するには、**CT\_MANAGEMENT\_SCOPE** を **2** に設定します。

**-r "resource\_handle"**  
リソース・ハンドルを指定します。リソース・ハンドルは、フォーマット : "**Oxnnnn Oxnnnn Oxnnnnnnnn Oxnnnnnnnn Oxnnnnnnnn Oxnnnnnnnn**" (ここで、**n** は任意の有効な 16 進数字) を使用して指定します。リソース・ハンドルは、一意的に除去の必要がある特定のリソース・インスタンスを識別します。

**-s "selection\_string"**  
選択文字列を指定します。選択文字列はすべて、二重または単一の引用符で囲まなければなりません。選択文字列の中に二重引用符が含まれている場合は、選択文字列全体を単一の引用符で囲んでください。以下に例を示します。

```
-s 'Name == "testing"'  
-s 'Name != "test"'
```

永続属性のみを選択文字列にリストすることができます。選択文字列を指定する方法については、「*RSCT: Administration Guide*」を参照してください。

- h** コマンドの使用ステートメント (使用法) を標準出力に書き込みます。
- T** コマンドのトレース・メッセージを標準エラーに書き込みます。これはソフトウェア・サービス担当者のみが使用します。
- V** コマンドの詳細メッセージを標準出力に書き込みます。

## パラメーター

### *resource\_class*

リソース・クラス名を指定します。選択文字列の基準に一致するこのリソース・クラスのリソース・インスタンスが除去されます。

## セキュリティ

**rmrsrc** を実行するユーザーは、**rmrsrc** に指定する *resource\_class* に書き込み権限が必要です。権限は、相手のシステムのアクセス制御リスト (ACL) ファイルに指定します。ACL ファイルとその変更方法について詳しくは、「*RSCT: Administration Guide*」を参照してください。

## 終了状況

- 0** コマンドが正常に実行されました。
- 1** RMC でエラーが起きました。
- 2** コマンド行インターフェース (CLI) スクリプトでエラーが発生しました。
- 3** コマンド行に間違ったフラグが指定されました。
- 4** コマンド行に不正なパラメーターが指定されました。
- 5** 不正なコマンド行入力に基づく RMC により、エラーが発生しました。
- 6** 選択文字列と一致するリソースが見つかりません。

## 環境変数

### **CT\_CONTACT**

**CT\_CONTACT** 環境変数をホスト名または IP アドレスに設定すると、コマンドは指定されたホストの RMC (リソース・モニターおよび制御) デーモンと連絡します。この環境変数を指定しないと、コマンドは、そのコマンドが実行されているローカル・システムの RMC デーモンと連絡します。コマンドにより表示または変更されるリソースまたはリソース・クラスは、接続が確立されたシステム上で探されます。

### **CT\_IP\_AUTHENT**

**CT\_IP\_AUTHENT** 環境変数が存在する場合、RMC デーモンは、IP ベース・ネットワーク認証を使用して、**CT\_CONTACT** 環境変数が設定されている IP アドレスで指定されたシステム上の RMC デーモンに連絡します。**CT\_IP\_AUTHENT** は、**CT\_CONTACT** が IP アドレスに設定されている場合にのみ意味を持ちます。ドメイン・ネーム・システム (DNS) サービスには依存しません。

## CT\_MANAGEMENT\_SCOPE

RMC デーモンがリソースおよびリソース・クラスをモニターし、制御するためのセッションに使用する管理有効範囲を指定します。管理有効範囲は、リソースおよびリソース・クラスをモニターし、制御する対象となる、可能なターゲット・ノードのセットを決定します。有効な値は以下のとおりです。

- 0 *local* 有効範囲を指定します。
- 1 *local* 有効範囲を指定します。
- 2 *peer domain* (ピア・ドメイン) 有効範囲を指定します。
- 3 *management domain* (管理ドメイン) 有効範囲を指定します。

この環境変数を設定しないと、*local* 有効範囲が使用されます。

## 実装上の固有な条件

このコマンドは、AIX 用の Reliable Scalable Cluster Technology (RSCT) ファイルセットの一部です。

## 標準出力

**-h** フラグが指定されている場合は、このコマンドの使用状況の説明文が標準出力に書き込まれます。

コマンド出力とすべての詳細メッセージは、標準出力に書き込まれます。

## 標準エラー

トレース・メッセージはすべて、標準エラーに書き込まれます。

## 例

- リソース・クラス **IBM.Host** から **c175n05** という名前のリソースを除去するには、次のように入力します。

```
rmrsrc -s 'Name == "c175n05"' IBM.Host
```

- リソース・ハンドル **"0x4017 0x0001 0x00000000 0x0069684c 0x0d52332b3 0xf3f54b45"** とリンクしたリソースを除去するには、次のように入力します。

```
rmrsrc -r "0x4017 0x0001 0x00000000 0x0069684c 0x0d52332b3 0xf3f54b45"
```

- /tmp/common/node\_file** ファイルを使用して、クラスター内のノードから **IBM.Foo** から **Test1** という名前のリソースを除去するには、次のように入力します。

```
# common node file
#
node1.ibm.com    main node
node2.ibm.com    main node
node4.ibm.com    backup node
node6.ibm.com    backup node
#
```

次のように入力します。

```
rmrsrc -s 'Name == "Test1"' -N /tmp/common/node_file IBM.Foo
```

## 位置

**/usr/sbin/rsct/bin/rmrsrc**

## 関連情報

資料:

- 「*CSM: Administration Guide*」(ノード・グループに関する情報)
- 「*CSM: Command and Technical Reference*」(**nodegrp** コマンドに関する情報)
- 「*RSCT: Administration Guide*」(RMC 操作と、式および選択ストリングの使い方に関する情報)

コマンド: **lsrsrc**、**mkrsrc**、**nodegrp**

情報ファイル: **rmccli** (RMC 関連コマンドに関する一般情報)

---

## rmsensor コマンド

### 目的

センサーを Resource Monitoring and Control (RMC) サブシステムから除去します。

### 構文

```
rmsensor [-a | -n host1 [, host2 ...] | -N { node_file | "-" } ] [-h] [-v | -V] sensor_name1 [sensor_name2 ...]
```

### 説明

**rmsensor** コマンドは、リソース・モニターおよび制御 (RMC) サブシステムの中のセンサー・リソース・クラスから、名前で指定したセンサーを除去します。

このセンサーをモニターしようとしている場合、モニター処理は停止されることとなりますが、モニター処理用に定義された **ERRM** リソースは除去されません。それらのリソースも除去するには、このセンサー用に使用されていたモニターしているリソースに対して、**rmcondition**、**rmresponse**、または **rmcondresp** のコマンドを使用します。

**rmsensor** コマンドは、どのノードでも実行されます。ドメイン内のすべてのノードで **rmsensor** を実行したい場合は、**-a** フラグを使用します。ドメイン内の一部のノードで **rmsensor** を実行したい場合は、**-n** フラグを使用します。

### フラグ

- a**     ドメイン内のすべてのノード上の、指定された名前と一致するセンサーを除去します。クラスターの有効範囲は、**CT\_MANAGEMENT\_SCOPE** 環境変数によって決まります。  
**CT\_MANAGEMENT\_SCOPE** が設定されていない場合、コマンドにとって有効な有効範囲になるまで、最初に管理ドメイン有効範囲 (存在する場合) を選択し、次にピア・ドメイン有効範囲 (存在する場合) を選択し、次にローカル有効範囲を選択します。コマンドは、最初に見つかった有効な有効範囲に対して、1 回実行されます。例えば、管理ドメインとピア・ドメインの両方が存在する場合に **CT\_MANAGEMENT\_SCOPE** を設定しない **rmsensor -a** を実行すると、それは管理ドメインで実行されます。この場合、ピア・ドメインで実行するには **CT\_MANAGEMENT\_SCOPE** を 2 に設定します。
- h**     コマンドの使用ステートメント (使用法) を標準出力に書き込みます。
- n *host1* [, *host2* ...]**  
センサーを除去するノードを指定します。デフォルトでは、センサーはローカル・ノードから除去されます。このフラグは、管理ドメインまたはピア・ドメインでのみ使用します。

**-N** {*node\_file* | "-"}  
センサーの削除対象となるノードをリストしているファイルまたは標準入力を指定します。このフラグの使用が適切なのは、クラスター・システム・マネージメント (CSM) またはピア・ドメイン・クラスターの中だけです。

**-v** | **-V**

コマンドの詳細メッセージを標準出力に書き込みます。

## パラメーター

*sensor\_name1* [*sensor\_name2...*]

除去する、1 つまたは複数のセンサーの名前を指定します。

## セキュリティ

**rmsensor** を実行するユーザーは、**IBM.Sensor** リソース・クラスに書き込み権限が必要です。権限は、相手のシステムのアクセス制御リスト (ACL) ファイルに指定します。ACL ファイルおよびその変更方法について詳しくは、「*RSCT: Administration Guide*」を参照してください。

## 終了状況

- 0 コマンドが正常に実行されました。
- 1 フラグとパラメーターの間違った組み合わせが入力されました。
- 6 センサー・リソースが見つかりませんでした。
- n* RMC サブシステムから戻される可能性のあるその他のエラーに基づくものです。

## 環境変数

### CT\_CONTACT

**CT\_CONTACT** 環境変数をホスト名または IP アドレスに設定すると、コマンドは指定されたホストの RMC (リソース・モニターおよび制御) デーモンと連絡します。この環境変数を設定しないと、コマンドは、そのコマンドが実行されているローカル・システムの RMC デーモンと連絡します。コマンドにより表示または変更されるリソースまたはリソース・クラスは、接続が確立されたシステム上で探されます。

### CT\_IP\_AUTHENT

**CT\_IP\_AUTHENT** 環境変数が存在する場合、RMC デーモンは、IP ベース・ネットワーク認証を使用して、**CT\_CONTACT** 環境変数が設定されている IP アドレスで指定されたシステム上の RMC デーモンに連絡します。**CT\_IP\_AUTHENT** は、**CT\_CONTACT** が IP アドレスに設定されている場合にのみ意味を持ちます。ドメイン・ネーム・システム (DNS) サービスには依存しません。

### CT\_MANAGEMENT\_SCOPE

RMC デーモンがリソースおよびリソース・クラスをモニターし、制御するためのセッションに使用する管理有効範囲を指定します。管理有効範囲は、リソースおよびリソース・クラスをモニターし、制御する対象となる、可能なターゲット・ノードのセットを決定します。

有効な値は以下のとおりです。

- 0 *local* 有効範囲を指定します。
- 1 *local* 有効範囲を指定します。



- 2 *peer domain* (ピア・ドメイン) 有効範囲を指定します。
- 3 *management domain* (管理ドメイン) 有効範囲を指定します。

この環境変数を設定しないと、*local* 有効範囲が使用されます。

## 実装上の固有な条件

このコマンドは、AIX 用の Reliable Scalable Cluster Technology (RSCT) ファイルセットの一部です。

## 例

1. **sensor1** というセンサーを除去するには、次のように入力します。

```
rmsensor sensor1
```

## 位置

`/usr/sbin/rsct/bin/rmsensor`

## 関連情報

コマンド: **chsensor**、**lssensor**、**mksensor**、**refsensor**、**rmcondition**、**rmresponse**、**rmcondresp**

資料: 「*RSCT: Administration Guide*」 (ACL 許可ファイルに関する情報)

---

## rmserver コマンド

### 目的

Subserver Type オブジェクト・クラスからサブサーバー定義を除去します。

### 構文

```
rmserver -t Type
```

### 説明

**rmserver** コマンドは、Subserver Type オブジェクト・クラスから既存のサブサーバー定義を除去します。

### フラグ

**-t *Type*** 除去する既存のサブサーバーを特定に識別するサブサーバー名を指定します。*Type* 名が Subserver Type オブジェクト・クラスで定義されていないと、**rmserver** コマンドは失敗します。

## セキュリティ

監査イベント: サブシステムの監査が正しく構成されていて使用可能にされると、**rmserver** コマンドを実行するたびに次のような監査レコード (イベント) が生成されます。

| イベント                 | 情報                            |
|----------------------|-------------------------------|
| <b>SRC_Delserver</b> | 削除されたサブサーバー定義の名前を監査ログにリストします。 |



監査イベントの選択とグループ化、および監査イベント・データ収集の構成方法についての詳細は、オペレーティング・システムおよびデバイスのマネージの監査の設定のセクションを参照してください。

## ファイル

`/etc/objrepos/SRCsubsvr`

SRC Subserver Configuration オブジェクト・クラスを指定します。

## 関連情報

オペレーティング・システムおよびデバイスのマネージのシステム・リソース・コントローラーのセクション。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」のプログラマーのためのシステム・リソース・コントローラー (SRC) の概要のセクション。

`auditpr` コマンド、`chserver` コマンド、`mkserver` コマンド、`startsrc` コマンド、`stopsrc` コマンド。

セキュリティの監査の概要のセクション。

---

## rmsock コマンド

### 目的

ファイル・ディスクリプターを持たないソケットを除去します。

### 構文

`rmsock Address TypeofAddress`

### 説明

`rmsock` コマンドは、ファイル・ディスクリプターを持たないソケットを除去します。このコマンドは、`socket`、`tcpcb`、`inpcb`、`ripcb`、または `rawcb` アドレスを受け入れ、それをソケット・アドレスに変換します。すべてのプロセス内のすべてのオープン・ファイルは、ソケットと一致しているかどうかを検出するために検査されます。一致しているファイルが見つかったら、ソケット `linger` オプションが指定されているかどうかに関係なく、そのソケットについてアボート・アクションが実行されます。ソケットが保持しているポート番号が解放されます。一致が見つかると、そのファイル・ディスクリプターとオーナー・プロセスの状況がユーザーに表示されます。その結果は `syslogd` にパスされ、`/var/adm/ras/rmsock.log` ファイルに記録されます。

除去対象のソケットがアクティブのプロセスによって保留されてはいないが、既存状態のプロセスが存在する場合は、`rmsock` は、指定されたソケットを除去しません。その理由は、そのソケットが既存状態のプロセスによって保留される可能性があるためです。既存プロセスによって保留されているソケットは、それらのプロセスが完全に終了した時点でクリーンアップされます。

### 例

1. ソケットをそのソケット・アドレスから除去するには、以下のように入力します。

```
rmsock 70054edc socket
```

ソケットの型を指定する必要はありません。ソケットの型には、 `tcpcb`、`udp`、`raw`、または `routing` ソケットがあります。

- ソケットをその `inpcb` アドレスから除去するには、以下のように入力します。

```
rmsock 70054edc inpcb
```

- ソケットをその `tcpcb` アドレスから除去するには、以下のように入力します。

```
rmsock 70054ecc tcpcb
```

## ファイル

`/usr/sbin`

`rmsock` コマンドが常駐しているディレクトリー。

`/var/adm/ras/rmsock.log`

`rmsock.log` ファイルが含まれています。

## 関連情報

`syslogd` デーモン。

---

## rmss コマンド

### 目的

アプリケーションのパフォーマンス・テストのために各種サイズのメモリーを使用してシステムをシミュレートします。

### 構文

```
rmss -c MemSize
```

```
rmss -r
```

```
rmss -p
```

```
rmss [ -d MemSize ] [ -f MemSize ] [ -n NumIterations ] [ -o OutputFile ] [ -s MemSize ] Command
```

### 説明

`rmss` コマンドはさまざまなサイズの実メモリーでシステムをシミュレートします。この場合、メモリー・ボードを取り外したり交換する必要はありません。さまざまなサイズのメモリーでアプリケーションを実行し、そのパフォーマンス統計情報を収集して、アプリケーションを適切なパフォーマンスで実行するのに必要なメモリーを決定します。 `rmss` コマンドを次の 2 つの目的で呼び出す場合は、以下のようにします。

- メモリー・サイズを変更して終了するには、 `-c`、`-p`、および `-r` フラグを使用します。このようにフラグを指定すると、ユーザーは任意のサイズのメモリーで自由にテストを行うことができます。
- このコマンドをドライバー・プログラムとして機能させるには、 `-s`、`-f`、`-d`、`-n`、および `-o` フラグを指定します。このモードでは、 `rmss` コマンドは指定したコマンドをメモリー・サイズの範囲を超えて複数回実行し、各メモリー・サイズでのコマンドのパフォーマンスを記述した重要な統計情報を表示します。このコマンドは、実行可能プログラムまたはシェル・スクリプト・ファイルを指定できます。また、コマンド行引数は指定してもしなくてもかまいません。

**重要:** **rmss** を複数メモリー・プール・システムで使用すると、以下のメッセージが出て失敗することがあります。

```
Failure: VMM unable to free enough frames for stealing.  
Choose a larger memory size or retry with less system activity.
```

あるいは、これと似たメッセージが出ることもあります。この失敗は、**rmss** がメモリー・プールからすべてのフレームをスチールし、他のプールからフレームをスチールできないときに発生することがあります。この解決策は、メモリーを増分単位で縮小することです。

システム上のメモリー・プールの数とサイズは、次のコマンドで検索できます。

```
echo "mempool *" | kdb
```

**-c**、**-p**、および **-r** フラグは相互に排他的です。**-c** フラグはメモリー・サイズを変更します。**-p** フラグは現在のメモリー・サイズを表示します。**-r** フラグはメモリー・サイズをコンピューターの実メモリー・サイズにリセットします。

**rmss** コマンドをドライバー・コマンドとして呼び出し、メモリー・サイズの範囲を超えて複数回実行するコマンド (実行可能プログラムまたはシェル・スクリプト・ファイル) のパフォーマンスを測定する場合は、**-s**、**-f**、**-d**、**-n**、**-o** フラグを組み合わせて指定します。このようにして **rmss** コマンドを呼び出すと、このコマンドは各メモリー・サイズでのパフォーマンス統計情報 (コマンドの応答時間やコマンドの実行中に発生したページ・インの回数など) を表示します。このような統計情報はファイルにも書き込まれません。これについてはこの例で説明します。

**-s** と **-f** フラグで範囲の開始点と終了点を指定し、**-d** フラグでこの範囲内でのメモリー・サイズの増分または減分を指定します。**-n** フラグを使用して各メモリー・サイズでのコマンドの実行回数を指定し、**-o** フラグを使用して **rmss** が生成するレポートの書き込み先となる出力ファイル名を指定します。Command パラメーターは、各メモリー・サイズで実行して (パフォーマンスを) 測定するコマンドを指定します。

注:

1. **rmss** コマンドは、“使用可能な”実メモリーを報告します。コンピューター上に不良メモリーがあったり、メモリーが使用中の場合、**rmss** は物理実メモリーから、不良メモリーまたはシステムが使用中のメモリーを差し引いた実メモリーの量を報告します。例えば、**-r** は **rmss** フラグを使用して以下のように表示することがあります。

```
Simulated Memory Size changed to 79.9062MB
```

このような結果になるのは、不良のマークを付けられたページがあるか、デバイスが使用のためにページをいくつかリザーブ (そのためユーザーが使用できない) しているためです。

2. デモンなどのバックグラウンド・プロセスと組み合わせられて実行されるアプリケーションが多くの個別ファイル (ディレクトリー・ファイルも含む) にアクセスする場合、**rmss** コマンドは、このアプリケーションの実行に必要なページ・インの回数を少なく見積もることがあります。このような過小評価が発生するのは、シミュレートされるメモリーのサイズが 8MB で、約 250 の個別ファイルにアクセスする場合です。任意のシミュレートされるメモリー・サイズでアクセスすると **rmss** コマンドがページ・インの要件を実際よりも少なく見積もる可能性のある個別ファイルのおおよその数を次の表に示します。

| シミュレートされるメモリー・サイズ (MB) | アクセスする個別ファイル |
|------------------------|--------------|
| 8                      | 250          |
| 16                     | 500          |
| 24                     | 750          |

| シミュレートされるメモリー・サイズ (MB) | アクセスする個別ファイル |
|------------------------|--------------|
| 32                     | 1000         |
| 48                     | 1500         |
| 64                     | 2000         |
| 128                    | 4000         |
| 256                    | 8000         |

あるコマンドが多く個々の個別ファイルにアクセスする可能性があると感じた場合には、**filemon** コマンドを使用して、該当するコマンドの実行時にアクセスされるファイルの数を判別することができます。

## フラグ

- c MemSize** シミュレートされるメモリーのサイズを *MemSize* の値に変更します。この値はメガバイト単位の整数または小数です。 *MemSize* 変数は 8MB からコンピューターの実メモリーのサイズの範囲で指定します。 **-c** フラグにはデフォルト値はありません。  
**注:** カーネルなどの内部システム構造のサイズとの関係から、シミュレートされるメモリーのサイズを 8MB 未満に変更することは困難です。
- d MemSize** シミュレート対象のメモリー・サイズ間の増分または減分を指定します。 *MemSize* 値はメガバイト単位の整数または小数です。 **-d** フラグを省略すると、増分または減分は 8MB になります。
- f MemSize** 最終メモリー・サイズを指定します。シミュレートされるメモリー・サイズ (*MemSize* 変数で指定される) でテスト・コマンドを実行して、シミュレートするシステムのテストを終了させてください。この変数はメガバイト単位の整数または小数です。 *MemSize* 変数は 4MB からコンピューターの実メモリーのサイズの範囲で指定します。 **-f** フラグを省略すると、最終メモリー・サイズは 8MB になります。  
**注:** カーネルなどの内部システム構造のサイズとの関係から、シミュレートされるメモリー・サイズが 8MB 未満で終了することは困難です。
- n NumIterations** 各メモリー・サイズでコマンドの実行と測定を行う回数を指定します。 **-n** フラグにはデフォルト値はありません。 **rmss** コマンドの初期化時に **-n** フラグを省略すると、 **rmss** コマンドは、10 秒間の合計実行回数を累算するのに必要なテスト・コマンドの反復回数を判別し、各メモリー・サイズでその回数だけコマンドを実行します。  
**注:** **rmss** コマンドは通常、実際に実行が測定される前に、テスト・コマンドを各メモリー・サイズで 1 回実行します。これにより、実際のテストのシミュレーションを行います。
- o OutputFile** **rmss** レポートの書き込み先となるファイルを指定します。 **-o** フラグを省略すると、 **rmss** レポートは **rmss.out** ファイルに書き込まれます。また、 **rmss** レポートは通常標準出力に書き出されます。
- p** シミュレートされるメモリー・サイズの現行値を表示します。
- r** シミュレートされるメモリー・サイズをコンピューターの実メモリーのサイズにリセットします。
- s MemSize** 開始メモリー・サイズを指定します。シミュレートされるメモリー・サイズ (*MemSize* 変数で指定される) でテスト・コマンドを実行して、シミュレートするシステムのテストを開始してください。この変数はメガバイト単位の整数または小数です。 *MemSize* 変数は 4MB からコンピューターの実メモリーのサイズの範囲で指定します。 **-s** フラグを省略すると、開始メモリー・サイズはコンピューターの実メモリーのサイズになります。  
**注:** カーネルなどの内部システム構造のサイズとの関係から、シミュレートされるメモリー・サイズが 8MB 未満で開始することは困難です。

*Command*

各メモリー・サイズで実行および測定するコマンドを指定します。 *Command* パラメーターには、実行可能プログラムまたはシェル・スクリプト・ファイルを指定できます。また、コマンド行引数は指定してもしなくてもかまいません。デフォルトのコマンドはありません。

## 終了状況

このコマンドは、以下の終了値を戻します。

0                    正常終了。  
>0                   エラーが発生しました。

## セキュリティ

アクセス制御: このコマンドを実行するには、root 権限を持っていないければなりません。

## 例

1. メモリー・サイズを 13.5MB に変更するには、次のように入力します。

```
rmss -c 13.5
```

2. 現在のメモリー・サイズを出力するには、次のように入力します。

```
rmss -p
```

3. メモリー・サイズをコンピューターの実メモリーのサイズにリセットするには、次のように入力します。

```
rmss -r
```

4. コマンド `cc -O foo.c` をメモリー・サイズ 32MB、24MB、16MB、8MB で実行し、そのパフォーマンスを各メモリー・サイズで測定して、レポートを `cc.rmss.out` ファイルに書き込むには、次のように入力します。

```
rmss -s 32 -f 8 -d 8 -n 1 -o cc.rmss.out cc -O foo.c
```

5. マシンの実メモリー・サイズから 8 MB までの範囲で 8MB ずつ増しながら、`foo.sh` シェル・スクリプト・ファイル内のコマンド・シーケンスのパフォーマンスを調べ、実行する反復の数の決定、および、それぞれのメモリー・サイズごとに行うファイルにおける `foo.sh` の測定を `rmss` コマンドで実行させてから、`rmss` レポートを `rmss.out` ファイルに (`rmss` コマンドのこの呼び出しで使用されるすべてのデフォルトを使って) 書き込むには、次のように入力します。

```
rmss foo.sh
```

6. メモリー・サイズを 8MB から 16MB の範囲で 0.5MB ずつ増加しながら、実行可能プログラム `bar` のパフォーマンスを測定し、各メモリー・サイズで `bar` を 2 回ずつ実行および測定し、レポートを `bar.rmss.out` ファイルに書き込むには、次のようにします。

```
rmss -s 8 -f 16 -d .5 -n 2 -o bar.rmss.out bar
```

7. `-s`、`-f`、`-d`、`-n`、`-o` フラグのいずれかを組み合わせて指定すると、`rmss` コマンドはドライバー・プログラムとして実行されます。ドライバー・プログラムは、メモリー・サイズの範囲を超えて複数回コマンドを実行して、各メモリー・サイズでのコマンドのパフォーマンスを記述して統計情報を表示します。

`rmss` コマンドによって出力されるレポートの例を以下に示します。

```
Hostname: xray.austin.ibm.com
Real memory size: 48.00 Mb
Time of day: Wed Aug 8 13:07:33 1990
```

```

Command: cc -O foo.c
Simulated memory size initialized to 24.00 Mb.
Number of iterations per memory size = 1 warmup + 1 measured = 2.
Memory size Avg. Pageins Avg. Response Time Avg. Pagein Rate
(megabytes) (sec.) (pageins/sec.)
-----
24.00 0.0 113.7 0.0
22.00 5.0 114.8 0.0
20.00 0.0 113.7 0.0
18.00 3.0 114.3 0.0
16.00 0.0 114.6 0.0
14.00 139.0 116.1 1.2
12.00 816.0 126.9 6.4
10.00 1246.0 135.7 9.2
8.00 2218.0 162.9 13.6

```

このレポートを生成するコマンドを以下に示します。

```
rmss -s 24 -f 8 -d 2 -n 1 cc -O foo.c
```

レポートの最初の部分は一般情報で、**rmss** コマンドが実行されていたコンピューター、そのコンピューターの実メモリーのサイズ、時刻と日付、測定されたコマンドなどが出力されます。その下の 2 行には、**rmss** コマンドの初期化について説明した情報メッセージが出力されます。上記の例では、**rmss** コマンドがシミュレートされるメモリー・サイズを 24MB に初期化していることを示しています。この 24MB は **-s** フラグで指定した開始メモリー・サイズです。また、**rmss** コマンドは、各メモリー・サイズでコマンドが実行される反復回数も印刷します。コマンドは立ち上げで 1 回、パフォーマンスの測定で 1 回の計 2 回ずつ、各メモリー・サイズで実行されます。反復回数は **-n** フラグで指定します。

レポートの後半の部分には、コマンドを実行した各メモリー・サイズについて、次のようなデータが出力されます。

- メモリー・サイズと、コマンドの実行時に発生したページ・インの平均数
- コマンドの平均応答時間
- コマンドの実行時に発生した平均ページ・イン率

注: ページ・インの平均数と平均ページ・イン率の値については、コマンドによって開始されたページ・インだけではなく、コマンドの実行時に発生したすべてのページ・インが対象となります。

## ファイル

`/usr/bin/rmss` **rmss** コマンドが入っています。

## 関連情報

**filemon** コマンド、および **svmon** コマンド。

---

## rmssys コマンド

### 目的

サブシステム・オブジェクト・クラスからサブシステム定義を除去します。



## 構文

**rmssys -s** *Subsystem*

## 説明

**rmssys** コマンドは、サブシステム・オブジェクト・クラスから既存のサブシステム定義を除去します。また、除去対象となるサブシステム用に存在するサブサーバーと通知メソッド定義も除去します。

## フラグ

**-s** *Subsystem*                    除去されるサブシステムを固有に識別する名前を指定します。サブシステム・オブジェクト・クラス内で既知のサブシステム名を指定しなければ、**rmssys** コマンドは異常終了します。  
**rmssys** コマンドは、このサブシステム用に定義されたサブサーバー定義を Subserver Type オブジェクト・クラスから除去し、このサブシステム用に定義された通知メソッド定義を通知オブジェクト・クラスから除去します。

## セキュリティ

監査イベント: 監査サブシステムが正しく構成され、使用可能になっていれば、**rmssys** コマンドは、コマンドが実行されるたびに下記の監査レコード (イベント) を生成します。

| イベント               | 情報                        |
|--------------------|---------------------------|
| <b>SRC_Delssys</b> | 除去されるサブシステムの名前の監査ログ内のリスト。 |

監査イベントの選択とグループ化、および監査イベント・データ収集の構成方法についての詳細は、セキュリティの監査の設定のセクションを参照してください。

## ファイル

|                                |                                               |
|--------------------------------|-----------------------------------------------|
| <b>/etc/objrepos/SRCsubsys</b> | SRC サブシステム構成オブジェクト・クラスを指定します。                 |
| <b>/etc/objrepos/SRCsubsvr</b> | SRC Subserver Configuration オブジェクト・クラスを指定します。 |
| <b>/etc/objrepos/SRCnotify</b> | SRC 通知メソッド・オブジェクト・クラスを指定します。                  |
| <b>/dev/SRC</b>                | <b>AF_UNIX</b> ソケット・ファイルを指定します。               |
| <b>/dev/.SRC-unix</b>          | 一時ソケット・ファイルの位置を指定します。                         |

## 関連情報

**auditpr** コマンド、**mkssys** コマンド、**lssrc** コマンド、**chssys** コマンド、**rmserver** コマンド、**refresh** コマンド、**startsrc** コマンド、**stopsrc** コマンド、**tracesoff** コマンド。

オペレーティング・システムおよびデバイスのマネージのシステム・リソース・コントローラーのセクション。

「AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグ」のプログラマーのためのシステム・リソース・コントローラー (SRC) の概要のセクション。

セキュリティの監査の概要のセクション。



---

## rmt コマンド

### 目的

磁気テープ・デバイスのリモート・アクセスを可能にします。

### 構文

rmt

### 説明

**rmt** コマンドを使用すると、磁気テープ・デバイスにリモート・アクセスすることができます。リモート・ダンプおよび復元を行うプログラムは、リモート磁気テープ・プロトコル・モジュールとして **rmt** コマンドを使用します。**rmt** コマンドは通常 **rexec** または **rcmd** サブルーチンによって開始されます。

**rmt** コマンドは、磁気テープの操作に特有の要求を受け入れて、コマンドを実行し、状況を示して応答します。応答はすべて ASCII 表記で、次の 2 つのフォーマットで行われます。正常コマンドは、Axxx という応答を受け取ります。ここで、xxx は 10 進数の ASCII 表記です。コマンドが異常終了すると、応答 Eyyy error-message を受け取ります。ここで、yyy は **errno.h** ファイルに記述されている可能なエラー番号の 1 つであり、error-message は **perror** サブルーチンのコールから印刷されるエラー文字列です。このプロトコルは以下のサブコマンドから構成されています。

### サブコマンド

|                                |                                                                                                                                                                                                                      |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>O</b> <i>DeviceMode</i>     | <i>Mode</i> パラメーターで指定したモードを使用して、 <i>Device</i> パラメーターで指定したデバイスをオープンします。 <i>Device</i> パラメーターの値はフル・パス名です。また、 <i>Mode</i> パラメーターの値は、 <b>open</b> サブルーチンに渡すのに適した、10 進数の ASCII 表記です。オープン・デバイスは、新しいオープン操作が行われる前にクローズされます。 |
| <b>C</b> <i>Device</i>         | オープン・デバイスをクローズします。 <i>Device</i> パラメーターで指定したデバイスは無視されます。                                                                                                                                                             |
| <b>L</b> <i>WhenceOffset</i>   | 指定されたパラメーターを使用して、 <b>lseek</b> 操作を行います。 <b>lseek</b> サブルーチンは応答値を戻します。                                                                                                                                                |
| <b>W</b> <i>Count</i>          | データをオープン・デバイスに書き込みます。 <b>rmt</b> コマンドは接続から、 <i>Count</i> パラメーターによって指定されたバイト数を読み取り、途中でファイル終わりがあると終了します。 <b>write</b> サブルーチンは応答値を戻します。                                                                                 |
| <b>R</b> <i>Count</i>          | <i>Count</i> パラメーターで指定したデータのバイト数をオープン・デバイスから読み取ります。次に、 <b>rmt</b> コマンドは要求された読み取り操作を行って、Azzz という値で応答します。ここで、zzz は、操作が成功した場合に読み込まれたバイト数です。読み取られたデータは次に送信されます。読み取り操作が失敗すると、標準フォーマットのエラーが戻されます。                          |
| <b>I</b> <i>OperationCount</i> | 指定したパラメーターを使用して、 <b>STIOCTOP</b> <b>ioctl</b> サブルーチンを実行します。パラメーターは 10 進数の ASCII 表記として解釈され、 <b>ioctl</b> サブルーチンで使われる構造の <i>mt op</i> および <i>mt count</i> フィールドに配置されます。操作が終了すると、 <i>Count</i> パラメーターの値が戻されます。          |

その他のサブコマンドを指定すると、**rmt** コマンドは終了します。

注: **R** および **W** サブコマンドでは、*Count* パラメーターで指定したバイト数が接続時に処理可能なバイト数を超過していると、データは処理可能なサイズに切り捨てられます。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0 正常終了。
- >0 エラーが発生しました。

## ファイル

`/usr/sbin/rmt` `rmt` コマンドが入っています。  
`/usr/include/sys/errno.h` 発生する可能性のあるエラー番号を記述します。

## 関連情報

`rdump` コマンド、 `rrestore` コマンド。

`rexec` サブルーチン。

`rmt` スペシャル・ファイル。

---

## rmtcpip コマンド

### 目的

ホスト・マシンの TCP/IP の構成を除去します。

### 構文

`rmtcpip`

### 説明

`rmtcpip` コマンドはホスト・マシン上の TCP/IP の構成を除去します。このコマンドの基本機能は以下のとおりです。

- ネットワーク・インターフェース構成を除去します。
- `/etc/rc.tcpip` をインストールされた初期状態に復元します。
- `/etc/hosts` をインストールされた初期状態に復元します。
- `/etc/resolv.conf` ファイルを除去します。
- デフォルトおよび静的経路を除去します。
- ホスト名をローカル・ホストに設定します。
- ホスト ID 127.0.0.1 にセットします。
- 構成データベースをインストールされた初期状態にリセットします。

注:

1. デフォルトで `/etc/rc.tcpip` にコメント化されたデーモンのうち、このコマンドを出した時点で実行中のデーモンが停止します。
2. `/etc/hosts` ファイルのユーザーのバージョンは、インストール時の状態に復元されている `etc/hosts` ファイルに保管される前に、 `/etc/hosts.save` に保管されます。

3. `/etc/resolv.conf` ファイルのバージョンが `/etc/resolv.conf.save` として保管されてから、`/etc/resolv.conf` ファイルが除去されます。

## セキュリティ

このコマンドは `root` からのみ実行できます。

## 関連情報

`hostent` コマンド、`hostname` コマンド、`hostid` コマンド、`mktcpip` コマンド。

`/etc/resolv.conf` ファイル・フォーマット。

---

## rmts コマンド

### 目的

シン・サーバーを除去します。

### 構文

```
rmts [-f] [-v] ThinServer
```

### 説明

`rmts` コマンドは、`ThinServer` によって指定され、`mkts` コマンドによって作成されたシン・サーバーを除去します。シン・サーバーが実行中の場合は、`rmts` コマンドはシン・サーバーを除去しません。その代わりに、シン・サーバーを除去できなかったことを示す、メッセージを印刷します。この場合、シン・サーバーのセッションを共通したイメージで終了させるには `-f` フラグを使用します。

### フラグ

|                 |                                             |
|-----------------|---------------------------------------------|
| <code>-f</code> | シン・サーバーが稼働中の場合、そのシン・サーバーを強制的に除去します。         |
| <code>-v</code> | <code>rmts</code> コマンドの実行時に詳細デバッグ出力を有効にします。 |

### 終了状況

|    |                  |
|----|------------------|
| 0  | コマンドは正常に実行されました。 |
| >0 | エラーが発生しました。      |

## セキュリティ

アクセス制御: `rmts` コマンドを実行するには、`root` 権限が必要です。

## 例

1. `lobo` という名前のシン・サーバーを除去するには、次のように入力します。

```
rmts lobo
```

## 位置

/usr/sbin/rmts

## ファイル

/etc/niminfo

NIM が使用する変数が入っています。

## 関連情報

**dbts** コマンド、**lsts** コマンド、**mkcosi** コマンド、**mkts** コマンド、**nim** コマンド、**nim\_clients\_setup** コマンド、**nim\_master\_setup** コマンド、**nimconfig** コマンド、**swts** コマンド。

---

## rmtun コマンド

### 目的

操作可能なトンネルを活動停止し、オプションでトンネルの定義を除去します。

### 構文

```
rmtun -v 4|6 -t tid_list | all [-d]
```

### 説明

**rmtun** コマンドは、アクティブなトンネル (複数の場合あり) を活動停止させ、オプションでトンネル定義 (複数の場合あり) を除去するのに使用します。また、このコマンドは、トンネル定義がトンネル・データベースから除去される時点で、**gentun** コマンドによってトンネルに対して作成された自動生成フィルター規則も除去します。

### フラグ

|                 |                                                                                                                                     |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <b>all</b>      | すべてのトンネルを活動停止し、オプションで除去します。                                                                                                         |
| <b>tid_list</b> | 活動停止したいトンネルのリスト。トンネル ID は、「,」または「-」で区切ることができます。「-」を使用して ID の範囲を指定できます。例えば、1、3、5 - 7 と指定すると、リストに 1、3、5、6、7 の 5 つのトンネル ID があることになります。 |
| <b>-d</b>       | これを指定すると、トンネルはトンネル・データベースから除去されます。これはオプションのフラグです。                                                                                   |
| <b>-t</b>       | 活動停止したいトンネルのリスト。 <b>-d</b> を指定すると、リスト内のすべてのトンネル定義もトンネル・データベースから除去されます。                                                              |
| <b>-v</b>       | トンネルの IP バージョン。IP バージョン 4 トンネルに対しては、 <b>4</b> の値を使用してください。IP バージョン 6 トンネルに対しては、 <b>6</b> の値を使用してください。                               |

## 関連情報

**chtun** コマンド、**exptun** コマンド、**gentun** コマンド、**imptun** コマンド、**lstun** コマンド、および **mktun** コマンド。

---

## rmuser コマンド

### 目的

ユーザー・アカウントを除去します。

### 構文

```
rmuser [ -R load_module ] [ -p ] Name
```

### 説明

**rmuser** コマンドは、*Name* パラメーターで指定されたユーザー・アカウントを除去します。ユーザーについて定義されたすべての属性が除去されますが、ユーザーのホーム・ディレクトリーとユーザーが所有するファイルは除去されません。ユーザー名は、既に存在してはなりません。 **-p** フラグを指定すると、**rmuser** コマンドは、**/etc/security/passwd** ファイルからパスワードおよび他のユーザーの認証情報も除去します。

代替 Identification and Authentication (I&A) メカニズムを使用して作成されたユーザーの場合は、該当するロード・モジュールを持つ **-R** フラグを使用してそのユーザーを除去しなければなりません。ロード・モジュールは、**/usr/lib/security/methods.cfg** ファイルに定義されます。

root ユーザー、または、UserAdmin 権限が与えられているユーザーだけが管理ユーザーを除去することができます。管理ユーザーは、**/etc/security/user** ファイルに **admin=true** と設定されているユーザーのことです。

Web-based System Manager (wsm) の ユーザー アプリケーションを使用してユーザー特性を変更できます。

このコマンドは、System Management Interface Tool (SMIT) **smit rmuser** 高速パスを使用して実行することもできます。

### フラグ

|                              |                                                       |
|------------------------------|-------------------------------------------------------|
| <b>-p</b>                    | <b>/etc/security/passwd</b> ファイルからユーザー・パスワード情報を除去します。 |
| <b>-R <i>load_module</i></b> | ユーザーの除去に使用するロード可能 I&A モジュールを指定します。                    |

### 終了状況

このコマンドは、以下の終了値を戻します。

|    |                                                        |
|----|--------------------------------------------------------|
| 0  | コマンドが正常に実行され、要求されたすべての変更が実行されます。                       |
| >0 | エラーが発生しました。プリントされたエラー・メッセージに、障害のタイプについての詳細情報が提供されています。 |

### セキュリティ

アクセス制御: このコマンドは、実行 (x) アクセス権を root ユーザーとセキュリティ・グループのメンバーだけに与えます。このコマンドは、トラステッド・コンピューティング・ベース (TCB) 内のプログラムとしてインストールします。このコマンドは、**setuid** (SUID) ビットが設定されている root ユーザーが所有しなければなりません。

アクセスされるファイルは次のとおりです。

| モード | ファイル                       |
|-----|----------------------------|
| rw  | /etc/passwd                |
| rw  | /etc/security/passwd       |
| rw  | /etc/security/user         |
| rw  | /etc/security/user.roles   |
| rw  | /etc/security/limits       |
| rw  | /etc/security/environ      |
| rw  | /etc/security/audit/config |
| rw  | /etc/group                 |
| rw  | /etc/security/group        |

監査イベントは次のとおりです。

| イベント        | 情報   |
|-------------|------|
| USER_Remove | ユーザー |

## 例

1. ユーザー `davis` のアカウントとその属性をローカル・システムから除去するには、次のように入力します。

```
rmuser davis
```

2. ユーザー `davis` のアカウントとそのすべての属性だけでなく、`/etc/security/passwd` ファイル内のパスワードや他のユーザー認証情報も除去するには、次のように入力します。

```
rmuser -p davis
```

3. LDAP ロード・モジュールによって作成されたユーザー `davis` を除去するには、次のように入力します。

```
rmuser -R LDAP davis
```

## ファイル

|                                         |                                  |
|-----------------------------------------|----------------------------------|
| <code>/usr/sbin/rmuser</code>           | <code>rmuser</code> コマンドが入っています。 |
| <code>/etc/passwd</code>                | ユーザーの基本属性が入っています。                |
| <code>/etc/security/passwd</code>       | パスワード情報が入っています。                  |
| <code>/etc/security/limits</code>       | 各ユーザーのリソース割り当てと制限を定義します。         |
| <code>/etc/security/user</code>         | ユーザーの拡張属性が入っています。                |
| <code>/etc/security/user.roles</code>   | ユーザーの管理ロール属性が入っています。             |
| <code>/etc/security/environ</code>      | ユーザーの環境属性が入っています。                |
| <code>/etc/security/audit/config</code> | 監査構成情報が入っています。                   |
| <code>/etc/group</code>                 | グループの基本属性が入っています。                |
| <code>/etc/security/group</code>        | グループの拡張属性が入っています。                |

## 関連情報

`chfn` コマンド、`chgrp` コマンド、`chsh` コマンド、`chgroup` コマンド、`chuser` コマンド、`ls` コマンド、`lsuser` コマンド、`mkg` コマンド、`mkuser` コマンド、`passwd` コマンド、`pwdadm` コマンド、`rmgroup` コマンド、`setgroups` コマンド、`setse` コマンド。

ユーザーの識別と認証、任意アクセス制御、トラステッド・コンピューティング・ベース、および監査の詳細については、セキュリティのネットワークの保護のセクションを参照してください。

管理ロールの詳細については、セキュリティのユーザー、ロール、およびパスワードのセクションを参照してください。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

---

## rmvfs コマンド

### 目的

**/etc/vfs** ファイル内のエントリーを除去します。

### 構文

**rmvfs** *VfsName*

### 説明

**rmvfs** コマンドは、**/etc/vfs** ファイルから *VfsName* エントリーを除去します。 *VfsName* パラメーターは仮想ファイルシステムの名前です。 **rmvfs** コマンドは、ファイルから除去する仮想ファイルシステム・タイプの名前を 1 つの引数としてとります。 *VfsName* エントリーが存在する場合は、ファイルから除去されます。

### 例

**newvfs** エントリーを除去するには、次のように入力します。

```
rmvfs newvfs
```

### ファイル

**/etc/vfs** 仮想ファイルシステム・タイプの記述が入っています。

### 関連情報

**chvfs** コマンド、 **crvfs** コマンド、 **lsvfs** コマンド、 **mount** コマンド。

オペレーティング・システムおよびデバイスのマネージのファイルシステムのセクションで、ファイルシステムのタイプ、管理、構造、および保守について説明しています。

---

## rmvirprt コマンド

### 目的

仮想プリンターを除去します。

### 構文

**rmvirprt -q** *PrinterQueueName* **-d** *QueueDeviceName*



## 説明

**rmvirprt** コマンドは、*PrinterQueueName* 変数の値と *QueueDeviceName* 変数の値に割り当てられている仮想プリンターを除去します。**rmvirprt** コマンドは、指定されたキューまたはキュー・デバイスに関連付けられている System Management Interface Tool (SMIT) オブジェクト・データ・マネージャー (ODM) オブジェクトも除去します。

Web-based System Manager のプリンター・キュー・アプリケーションを使用して、プリンター特性を変更することができます。

このコマンドは、System Management Interface Tool (SMIT) **smit rmvirprt** 高速パスを使用して実行することもできます。

**注:** コマンド **rmvirprt** は、コマンド行から実行されると、キューやキュー・デバイスの除去も、実行中のジョブ、または指定したキューやキュー・デバイスに入っているジョブの検査も実行しません。ただし、SMIT を使用してこのコマンドを対話式に実行する場合には、実行中のジョブまたはキューに入っているジョブがない限り、対応するキュー、キュー・デバイス、およびオプションでプリンター・デバイスが仮想プリンターとともに除去されます。

## フラグ

**-d** *QueueDeviceName* 仮想プリンターが割り当てられているキュー・デバイスの名前を指定します。  
**-q** *PrinterQueueName* 仮想プリンターが割り当てられている印刷キュー名を指定します。

## 例

proq 印刷キューに関連付けられている mypro 仮想プリンターの属性値を除去するには、下記のように入力します。

```
rmvirprt -d mypro -q proq
```

## ファイル

**/etc/qconfig** 構成ファイルが入っています。  
**/usr/sbin/rmvirprt** **rmvirprt** コマンドが入っています。  
**/var/spool/lpd/pio/@local/custom/\*** カスタマイズされた仮想プリンターの属性ファイルが入っています。  
**/var/spool/lpd/pio/@local/ddi/\*** ソートした仮想プリンターの属性ファイルが入っています。

## 関連情報

**chvirprt** コマンド、**lsvirprt** コマンド、**mkvirprt** コマンド、**smit** コマンド。

プリンターおよび印刷 の 印刷の管理 のセクション。

プリンターおよび印刷 の印刷スプーラーのセクション。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

---

## rmyp コマンド

### 目的

NIS の構成を除去します。

### 構文

```
/usr/sbin/rmyp { -s | -c }
```

### 説明

**rmyp** コマンドは、NIS 作業を行うために使用するシステムからすべてのものを除去します。例えば、**rmyp** コマンドは、すべての NIS マップおよび **/etc/rc.nfs** ファイル内の NIS デーモンに関するすべてのエントリを除去します。

Web-based System Manager (wsm) のネットワーク・アプリケーションを使用してネットワーク特性を変更できます。

このコマンドは、System Management Interface Tool (SMIT) **smit rmyp** 高速パスを使用して実行できます。このコマンドは、System Management Interface Tool (SMIT) を使用して実行できます。SMIT を使用するには、以下のように入力します。

```
smit rmyp
```

### フラグ

- s** システムからサーバー構成を除去します。
- c** システムからクライアント構成を除去します。

### 関連情報

**chslave** コマンド、**mkclient** コマンド、**mkslave** コマンド、**smit** コマンド。

**ypbind** デーモン、**yppasswd** デーモン、**ypserv** デーモン、**ypupdated** デーモン。

ネットワークおよびコミュニケーションのマネージのネットワーク・ファイルシステム (NFS) の概要のセクション。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

オペレーティング・システムおよびデバイスのマネージのSystem Management Interface Toolのセクション。

*AIX 5L Version 5.3 Network Information Services (NIS and NIS+) Guide*の Network Information Service (NIS) のセクション。

NIS のリファレンス。

---

## rndc コマンド

### 目的

ネーム・サーバー制御ユーティリティー。

### 構文

```
rndc [-c config-file] [-k key-file] [-s server] [-p port] [-V] [-y key_id] [command]
```

### 説明

**rndc** はネーム・サーバーの操作を制御します。これは、古い BIND リリースで提供された **ndc** ユーティリティーを置き換えるものです。コマンド行オプションまたは引数を指定せずに **rndc** を呼び出すと、サポートされるコマンドと使用可能オプションおよびその引数の簡易要約が印刷されます。

**rndc** は TCP 接続を経由してネーム・サーバーと通信し、デジタル署名を使用して認証されたコマンドを送信します。**rndc** および **named** の現行バージョンでは、サポートされる認証アルゴリズムは HMAC-MD5 だけであり、これは接続の両端で共有秘密を使用します。これは、コマンド要求およびネーム・サーバーの応答に TSIG スタイルの認証を提供します。チャンネル経由で送信されるすべてのコマンドには、サーバーが知っている **key\_id** により署名する必要があります。

**rndc** は構成ファイルを読み取って、ネーム・サーバーに連絡する方法を決定し、どのアルゴリズムおよびキーを使用すべきかを決めます。

### フラグ

|                              |                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-c</b> <i>config-file</i> | デフォルトの <b>/etc/rndc.conf</b> ではなく、 <i>config-file</i> を構成ファイルとして使用します。                                                                                                                                                                                                                                                                                                                                                    |
| <b>-k</b> <i>key-file</i>    | デフォルトの <b>/etc/rndc.key</b> ではなく、 <b>key-file</b> をキー・ファイルとして使用します。<br><i>config-file</i> が存在しない場合、 <b>/etc/rndc.key</b> 内のキーを使用して、サーバーに送信されるコマンドを認証します。                                                                                                                                                                                                                                                                  |
| <b>-s</b> <i>server</i>      | <b>server</b> は、 <b>rndc</b> の構成ファイル内の <b>server</b> ステートメントと一致するサーバーの名前またはアドレスです。コマンド行にサーバーを指定しないと、構成ファイルのオプション・ステートメント内の <b>default-server</b> 文節で指定されたホストが使用されます。                                                                                                                                                                                                                                                      |
| <b>-p</b> <i>port</i>        | BIND 9 のデフォルトの制御チャンネル・ポート 953 ではなく、TCP ポート・ポートにコマンドを送信します。                                                                                                                                                                                                                                                                                                                                                                |
| <b>-V</b>                    | 詳細ロギングを使用可能にします。                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>-y</b> <i>keyid</i>       | 構成ファイルからのキー <b>keyid</b> を使用します。制御メッセージ妥当性検査を成功させるには、 <b>keyid</b> は、同じアルゴリズムおよび秘密文字列を使用して <b>named</b> に知られている必要があります。 <b>keyid</b> を指定しないと、 <b>rndc</b> は最初に、使用するサーバーの <b>server</b> ステートメント内で <b>key</b> 文節を探し、またはそのホストに <b>server</b> ステートメントがない場合は、 <b>options</b> ステートメントの <b>default-key</b> 文節を探します。構成ファイルには、認証された制御コマンドをネーム・サーバーに送信するために使用される共有秘密が含まれることに注意してください。したがって、汎用読み取りアクセス権または汎用書き込みアクセス権を持たせるべきではありません。 |

**rndc** がサポートするコマンドの完全なセットについては、「BIND 9 Administrator Reference Manual」を参照するか、または **rndc** を引数なしで実行し、そのヘルプ・メッセージを参照してください。

## 制約事項

**rndc** コマンドは **named9** デーモンとだけ働きます。 **key\_id** 用の共有秘密を、構成ファイルを使用せずに提供することはできません。

## 関連情報

**rndc.conf(5)**、**named8**、**named.conf(5)** および **ndc(8)** デーモン。

「BIND 9 Administrator Reference Manual」。

---

## rndc-confgen コマンド

### 目的

**rndc** コマンドの構成ファイルを生成します。

### 構文

```
rndc-confgen [ -a ] [ -b keysize ] [ -c keyfile ] [ -h ] [ -k keyname ] [ -p port ] [ -r randomfile ] [ -s address ]
```

### 説明

**rndc-confgen** コマンドは、**rndc** コマンドの構成ファイルを生成します。これは、**rndc.conf** ファイルおよび対応する制御ステートメントとキー・ステートメントを **named.conf** に手作業で書き込まなくても済む、便利な代替方法として使用することができます。**rndc-confgen** コマンドに **-a** フラグを指定して実行し、**rndc.key** ファイルをセットアップすることができます。これを実行すれば、**rndc.conf** ファイルと制御ステートメントの必要はありません。

注: **rndc-confgen** コマンドは、**named9** デーモンとのみ働きます。

### フラグ

- a** 自動 **rndc** 構成を実行します。これは、始動時に **rndc** と **named** の両方が読み取る **/etc** (または BIND 作成時に **sysconfdir** に指定されたロケーション) に、ファイル **rndc.key** を作成します。**rndc.key** ファイルは、デフォルトのコマンド・チャンネルおよび、**rndc** がそれ以上の構成を必要とせずに **named** と通信できるようにする認証キーを定義します。
- b** *keysize* 認証キーのサイズをビット単位で指定します。サイズは 1 から 512 ビットの間にする必要があります。デフォルトは 128 です。
- c** *keyfile* **-a** フラグと一緒に使用し、**rndc.key** の代わりにロケーションを指定します。
- h** **rndc-confgen** へのオプションと引数の簡易要約を印刷します。
- k** *keyname* **rndc** 認証キーのキー名を指定します。これは有効なドメイン名でなければなりません。デフォルトは **rndc-key** です。
- p** *port* **named** が **rndc** からの接続を listen する、コマンド・チャンネル・ポートを指定します。デフォルトは 953 です。
- r** *randomfile* 許可を生成するためのランダム・データのソースを指定します。オペレーティング・システムが **/dev/random** または同等のデバイスを提供しない場合は、ランダム性のデフォルトのソースはキーボード入力です。**randomdev** には、デフォルトの代わりに使用されるランダム・データを含む、キャラクター・デバイスまたはファイルの名前を指定します。特殊値のキーボードはキーボード入力を使用することを示します。
- s** *address* **named** が **rndc** からのコマンド・チャンネル接続を listen する、IP アドレスを指定します。デフォルトは、ループバック・アドレス 127.0.0.1 です。

## 例

1. **rndc** を手作業による構成をせずに使用するには、次のコマンドを実行します。

```
rndc-confgen -a
```

2. **named.conf** に手作業で挿入するため、サンプルの **rndc.conf** ファイルおよび対応する制御ステートメントとキー・ステートメントを印刷するには、以下のコマンドを実行します。

```
rndc-confgen
```

## 関連情報

864 ページの『**rndc** コマンド』 および 1 ページの『**named** デーモン』。

---

## roffbib コマンド

### 目的

参考文献データベースを出力します。

### 構文

```
roffbib [ -m Macro ] [ -x ] [ FormatFlags ] [ Database... ]
```

### 説明

**roffbib** コマンドは、脚注用または巻末の注用のフォーマットではなく参考文献データベース・フォーマットのレコードをすべて出力します。一般に、このコマンドは **sortbib** コマンドのフィルターとして使用します。

**roffbib** コマンドは、**nroff** コマンドと **troff** コマンドで使用するフラグのほとんど、特に **-e**、**-h**、**-n**、**-o**、**-r**、**-s**、および **-T** フラグを受け入れます。

**%X** キー・フィールドに続けて要約または注釈を入力すると、注釈付き参考文献一覧用のパラグラフにフォーマットされます。複数の注釈付きパラグラフを希望する場合は、複数の **%X** フィールドを使うことができます。

### パラメーター

|                    |                                                                                                                                |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------|
| <i>FormatFlags</i> | <b>nroff</b> コマンド・フラグのほとんど、特に <b>-e</b> 、 <b>-h</b> 、 <b>-n</b> 、 <b>-o</b> 、 <b>-r</b> 、 <b>-s</b> 、および <b>-T</b> フラグを受け入れます。 |
| <i>Database</i>    | 全レコードの参考文献データベースを格納します。                                                                                                        |

### フラグ

|                        |                                                                                                                                                                                   |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-m</b> <i>Macro</i> | ユーザー定義のマクロ・セットが入っているファイルを指定します。 <b>-m</b> フラグとマクロの間にはスペースを 1 つ入れなければなりません。このマクロ・セットは <b>/usr/share/lib/tmac/tmac.bib</b> ファイル内で定義されたマクロを置換します。ユーザーはマクロを書き直してカスタマイズ・フォーマットを作成できます。 |
| <b>-x</b>              | <b>%X</b> フィールド・キーに続けて入力された要約またはコメントの印刷を抑制します。                                                                                                                                    |

## 例

**roffbib** コマンドを **sortbib** コマンドと一緒に使用したときの例を以下に示します。

```
sortbib Database | roffbib
```

## ファイル

`/usr/share/lib/tmac/tmac.bib` **nroff** および **troff** コマンドが使用するマクロが入っています。

## 関連情報

**addbib** コマンド、 **indxbib** コマンド、 **lookbib** コマンド、 **nroff** コマンド、 **refer** コマンド、 **sortbib** コマンド、 **troff** コマンド。

---

## rollback コマンド

### 目的

JFS2 ファイルシステムをポイント・イン・タイム・スナップショットに戻します。

### 構文

```
rollback [-s ] [ -v ] [-c] snappedFS snapshotObject
```

### 説明

**rollback** コマンドは、JFS2 ファイルシステムをポイント・イン・タイム・スナップショットに戻すインターフェースです。 *snappedFS* パラメーターは、**rollback** 実行前にアンマウントする必要があり、このコマンドの実行中はアクセス不能にされます。 *snapshotObject* のあとにとられたスナップショットはすべて、関連する論理ボリュームとともに除去されます。

**rollback** コマンドが何らかの理由で中断された場合は、このコマンドが再始動されて完了するまで、*snappedFS* パラメーターはアクセス不能にされます。再始動された **rollback** は、その初期コマンドと同じ *snapshotObject* をターゲットにする必要があります。

### フラグ

- |           |                                                                                                                                                |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-c</b> | このフラグを指定した場合は、読み取りまたは書き込みエラーがスナップショットからの <i>snappedFS</i> の復元時に監視される場合でも、 <b>rollback</b> は続行されます。デフォルトの動作は、これらのエラーの発生時にエラー・メッセージを出し、停止することです。 |
| <b>-s</b> | このフラグが指定された場合は、 <b>rollback</b> によって除去されるスナップショットに関連した論理ボリュームはすべて、保存されます。スナップショットは引き続き削除されます。                                                  |
| <b>-v</b> | これは詳細オプションであり、これにより、復元されたブロック・カウントがロールバック進行時に印刷されます。                                                                                           |

### パラメーター

*snappedFS*  
*snapshotObject*

ロールバックする JFS2 システム。  
戻される元のスナップショットの論理ボリューム。

## 例

**/home/janet/sb** ファイルシステムを論理ボリューム **/dev/snapsb** 上のスナップショットにロールバックするには、次のように入力します。

```
rollback /home/janet/sb /dev/snapsb
```

## 位置

**/usr/sbin/rollback**

**rollback** コマンドが入っています。

## 関連情報

backsnap コマンド、snapshot コマンド。

---

## route コマンド

### 目的

経路指定テーブルを手動で操作します。

### 構文

```
route [ -f ] [ -n ] [ -q ] [ -C ] [ -v ] Command [ Family ] [ [ -net | -host ] Destination [ -prefixlen n ] [ -netmask [ Address ] ] Gateway ] [ Arguments ]
```

### 説明

**route** コマンドを使用すると、ネットワーク経路指定テーブルに手動で入力を行うことができます。**route** コマンドは、*Destination* 変数のネットワーク・アドレスを解釈して、ホストへの経路とネットワークへの経路を区別します。このネットワーク・アドレスは、シンボル名か数値アドレスのいずれかで指定できます。**route** コマンドは、**/etc/hosts** ファイルまたはネットワーク・ネーム・サーバーを使用して、すべてのシンボル名をアドレスに変えます。

特定のホストへの経路は、宛先に関連付けられた IP アドレスを解釈することでネットワークへの経路と区別されます。オプションのキーワード **-net** と **-host** を指定すると、宛先は強制的にそれぞれネットワークまたはホストとして解釈されます。宛先のローカル・アドレスに **INADDR\_ANY** が入っているか、または宛先がネットワークのシンボル名の場合は、ネットワークへの経路が想定されます。それ以外の場合は、ホストへの経路が想定されます。

例えば、**128.32** は **-host 128.0.0.32** と解釈され、**128.32.130** は **-host 128.32.0.130** と解釈され、**-net 128.32** は **128.32.0.0** と解釈され、**-net 128.32.130** は **128.32.130.0** と解釈されます。

経路がゲートウェイを通じてではなくインターフェースによる場合、**-interface** 引数を指定しなければなりません。指定したゲートウェイは共通ネットワークのホストのアドレスであり、送信に使用されるインターフェースを示します。

**-netmask** 引数はアドレス・パラメーターの前に指定してください (このパラメーターがネットワーク・マスクとして解釈されます)。このオプションが *Destination* パラメーターの後に指定されていることを確認して、**-inet** ケースで生成された暗黙のネットワーク・マスクをオーバーライドすることができます。



宛先またはゲートウェイに指定されたすべてのシンボル名は最初に、**gethostbyname** サブルーチンによってホスト名として調べられます。これに失敗すると、**getnetbyname** サブルーチンは、ネットワーク名として解釈されます。

**注:** 経路指定には、経路指定ソケットと新規メッセージ・タイプ RTM\_ADD、RTM\_DELETE、および RTM\_CHANGE が使用されます。このため、経路指定テーブルを修正できるのは root ユーザーだけです。

**flush** または **-f** コマンドを指定すると、経路指定はすべてのゲートウェイ・エントリーの経路指定テーブルを「フラッシュ」、すなわちクリアします。アドレス・ファミリーを記述したオプションのキーワードを指定して、指定のアドレス・ファミリーに宛先のある経路だけをフラッシュすることができます。

**netstat -r** コマンドは、経路指定テーブルに入っている現在のルーティング情報を表示します。

## フラグ

|                     |                                                                                   |
|---------------------|-----------------------------------------------------------------------------------|
| <b>-f</b>           | 経路指定テーブルの中のネットワーク・インターフェースに関連していないエントリー項目をすべて除去します。                               |
| <b>-n</b>           | フラッシュまたは他のアクションの結果を詳細モードで報告するときに、記号ではなく数値でホスト名とネットワーク名を表示します。                     |
| <b>-q</b>           | 抑止モードを指定し、すべての出力を抑制します。                                                           |
| <b>-C</b>           | 経路の追加と削除のためのルーティング・メッセージ全体にわたって、 <b>ioctl</b> 呼び出しのための優先を指定します。                   |
| <b>-v</b>           | 詳細モードを指定し、追加の詳細を印刷します。                                                            |
| <b>-net</b>         | <i>Destination</i> パラメーターをネットワークとして解釈することを示します。                                   |
| <b>-netmask</b>     | 宛先アドレスへのネットワーク・マスクを指定します。このオプションが <i>Destination</i> パラメーターの後に指定されていることを確認してください。 |
| <b>-host</b>        | <i>Destination</i> パラメーターをホストとして解釈することを示します。                                      |
| <b>-prefixlen n</b> | 宛先接頭部の長さ (ネットマスク内のビット数) を指定します。                                                   |

デフォルトの経路は、ホスト (ネットワーク上の単一のコンピューター) です。**-net** パラメーターまたは **-host** パラメーターを指定せずにアドレスのネットワーク部分を指定すると、ネットワークへの経路であると想定されます。アドレスのホスト部分は 0 です。

## パラメーター

|                    |                                                                          |
|--------------------|--------------------------------------------------------------------------|
| <i>Arguments</i>   | 下記から 1 つまたは複数の引数を指定します。引数の変数として <i>n</i> を指定する場合は、変数 <i>n</i> の値は正の整数です。 |
| <b>-active_dgd</b> | 経路上のアクティブ・デッド・ゲートウェイ検出を使用可能にします。                                         |
| <b>-cloning</b>    | 新しい経路のクローンを複製します。                                                        |
| <b>-genmask</b>    | TSEL の長さを抽出します。これはクローン経路を生成するときに使用されます。                                  |
| <b>-interface</b>  | インターフェースの経路指定エントリーを操作します。                                                |
| <b>-rtt n</b>      | 往復時間を指定します。                                                              |
| <b>-rttvar n</b>   | 往復時間の差異を指定します。                                                           |

**-sendpipe** *n*

送信ウィンドウのサイズを指定します。

**-recvpipe** *n*

受信ウィンドウのサイズを指定します。

**-allowgroup** *gid*

経路の使用を許可されるグループ ID を指定します。グループ ID は、使用を許可されたグループのリストに追加されるか、または使用を許可されないグループのリストから削除されます。

**-denygroup** *gid*

経路の使用を許可されないグループ ID を指定します。グループ ID は、使用を許可されないグループのリストに追加されるか、または使用を許可されたグループのリストから削除されます。

**-stopsearch**

経路指定テーブルの検索で経路が一致した場合には検索を停止しますが、グループ経路指定の制限により、この経路を使用することはできません。

**-mtu** *n* この経路の最大伝送単位を指定します。インターフェースの最大伝送単位を超えない範囲で TCP アプリケーションのインターフェース最大伝送単位をオーバーライドします。このフラグは、UDP を使用したアプリケーションの最大伝送単位には影響ありません。

**-hopcount** *n*

経路のゲートウェイの最大数を指定します。

**-policy** *n*

Multipath Routing に使用されるポリシーを指定します。*n* は 1 から 5 までの数値で、ここでこれらの数値は次のような意味になります。

1. 重み付けラウンドロビン
2. ランダム
3. 重み付けランダム
4. 最低使用状況
5. ハッシュ・ベース

ポリシーが明示的に設定されておらず、マルチパス経路指定が使用される場合、**mpr\_policy** と呼ばれるグローバルな **no** コマンド・オプションによって、使用されるポリシーが決まります。デフォルト・ポリシーは重み付けラウンドロビンで、重みがすべて 1 である場合にはラウンドロビンと同じように動作します。デフォルト・ポリシーが重み付けラウンドロビンであっても、ポリシーが設定されていない場合は、ネットワーク・オプション **mpr\_policy** が優先されます。これに対して、ポリシーが明示的に WRR (重み付けラウンドロビン) に設定されている場合は、この設定は **mpr\_policy** 設定をオーバーライドします。これらのポリシーの詳細については、**no** コマンドを参照してください。

**-weight** *n*

Multipath Routing 機能で重み付けポリシーとして使用される経路の重みを指定します。

|             |                            |                                                                                                                       |
|-------------|----------------------------|-----------------------------------------------------------------------------------------------------------------------|
|             | <b>-expire</b> <i>n</i>    | 経路指定プロトコルが使用する満了メトリックを指定します。                                                                                          |
|             | <b>-sssthresh</b> <i>n</i> | アウトバウンド・ゲートウェイ・バッファの制限を指定します。                                                                                         |
|             | <b>-lock</b>               | メトリック修飾子を個別にロックできるメタ修飾子を指定します。 <b>-lock</b> メタ修飾子は、ロックする各修飾子の前に指定してください。                                              |
|             | <b>-lockrest</b>           | 後続のメトリックをすべてロックできるメタ修飾子を指定します。                                                                                        |
|             | <b>-if</b> <i>ifname</i>   | この経路と関連付けるインターフェース (en0, tr0 ...) を指定します。こうすることにより、この経路が選択されたときに、このインターフェースを使ってパケットが送信されるようになります。                     |
|             | <b>-xresolve</b>           | 使用時にメッセージを出力します (外部ルックアップの場合)。                                                                                        |
|             | <b>-iface</b>              | 宛先が直接到達可能であることを示します。                                                                                                  |
|             | <b>-static</b>             | 手動で追加した経路を指定します。                                                                                                      |
|             | <b>-nostatic</b>           | カーネルまたはデーモンが追加した仮の経路を指定します。                                                                                           |
|             | <b>-reject</b>             | 一致した場合、未到達の ICMP を出力します。                                                                                              |
|             | <b>-blackhole</b>          | 更新中にパケットを通知なしで廃棄します。                                                                                                  |
|             | <b>-proto1</b>             | プロトコルを特定の経路指定フラグ番号 1 に設定します。                                                                                          |
|             | <b>-proto2</b>             | プロトコルを特定の経路指定フラグ番号 2 に設定します。                                                                                          |
| Command     |                            | 次の 6 つの選択肢の 1 つを指定します。                                                                                                |
|             | <b>add</b>                 | 経路を追加します。                                                                                                             |
|             | <b>flush</b> または <b>-f</b> | すべての経路を除去します。                                                                                                         |
|             | <b>delete</b>              | 特定の経路を削除します。                                                                                                          |
|             | <b>change</b>              | 経路のアスペクト (経路のゲートウェイなど) を変更します。                                                                                        |
|             | <b>monitor</b>             | ルーティング情報ベースの変更、経路指定ロックアップ・ミス、または問題のありそうなネットワーク区画化を報告します。                                                              |
|             | <b>get</b>                 | 宛先の経路指定を検索し、表示します。                                                                                                    |
|             | <b>set</b>                 | 経路のポリシーおよび重み属性を設定します。                                                                                                 |
| Family      |                            | アドレス・ファミリーを指定します。 <b>-inet</b> アドレス・ファミリーがデフォルトです。 <b>-inet6</b> ファミリーは、すべての後続アドレスが <b>inet6</b> ファミリーに入っていることを指定します。 |
| Destination |                            | 経路の宛先となるホストまたはネットワークを指定します。 <b>Destination</b> パラメーターは、シンボル名または数値アドレスで指定できます。                                         |
| Gateway     |                            | パケットのアドレス先のゲートウェイを指定します。 <b>Gateway</b> パラメーターは、シンボル名または数値アドレスで指定できます。                                                |

## 例

1. あるネットワーク上のコンピューターが別のネットワーク上のコンピューターにメッセージを送信できるように経路を設定するには、次のように入力します。

```
route add 192.100.201.7 192.100.13.7
```

192.100.201.7 アドレスは、受信コンピューター (*Destination* パラメーター) のアドレスです。

192.100.13.7 アドレスは、経路指定コンピューター (*Gateway* パラメーター) のアドレスです。

2. 経路を確立して特定のネットワーク上のユーザーにメッセージを送信できるようにするには、次のように入力します。

```
route add -net 192.100.201.0 192.100.13.7
```

192.100.201.0 アドレスは受信ネットワークのアドレス (*Destination* パラメーター) です。

192.100.13.7 アドレスは経路指定ネットワークのアドレス (*Gateway* パラメーター) です。

3. デフォルトのゲートウェイを確立するには、次のように入力します。

```
route add 0 192.100.13.7
```

*Destination* パラメーターに値 0 またはデフォルトのキーワードを指定すると、前もって定義されておらず、ネットワークに直接接続していない宛先に送信されるパケットは、すべてデフォルトのゲートウェイを経由することになります。 192.100.13.7 アドレスは、デフォルトとして選択されたゲートウェイのアドレスです。

4. ホスト・ゲートウェイ・テーブルをクリアするには、次のように入力します。

```
route -f
```

5. 重みとポリシー情報を指定して経路を追加するには、次のように入力します。

```
route add 192.158.2.2 192.158.2.5 -weight 5 -policy 4
```

6. 既に存在している経路の重みとポリシー属性を設定するには、次のように入力します。

```
route set 192.158.2.2 192.158.2.5 -weight 3 -policy
```

## 関連情報

**netstat** コマンド。

**routed** デーモン。

**gethostbyname** サブルーチン、**getnetbyname** サブルーチン。

**/etc/hosts** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ の TCP/IP の経路指定、TCP/IP の経路指定ゲートウェイ、TCP/IP のアドレッシングのセクション。

---

## routed デーモン

### 目的

ネットワーク経路指定テーブルを管理します。

## 構文

注: コマンド行から **routed** デーモンを制御するには、**SRC** コマンドを使用します。それぞれのシステムの始動時にこのデーモンを開始する場合には、**rc.tcpip** ファイルを使用します。

```
/usr/sbin/routed [ -d ] [ -g ] [ -t ] { -s | -q } [ LogFile]
```

## 説明

**/usr/sbin/routed** デーモンは、カーネルのネットワーク経路指定テーブルを管理します。すべての TCP/IP ゲートウェイ・プロトコルをサポートする **gated** デーモンとは違って、**routed** デーモンはルーティング情報プロトコル (RIP) しか適用しません。外部ゲートウェイ・プロトコル (EGP)、シンプル・ネットワーク管理プロトコル (SNMP)、または分散コンピューター・ネットワーク・ローカル・ネットワーク・プロトコル (HELLO) の経路指定が必要な場合には、**routed** デーモンを使用してはなりません。これらのプロトコルには、**gated** デーモンを使用してください。

注: 同じホスト上で **gated** デーモンと **routed** デーモンを稼働すると、予想できない事態が発生することがあります。

**routed** デーモンは、システムに構成されており、**up** としてマークされている直接接続されているホストおよびネットワークへのインターフェースを検出します。( **ifconfig** コマンドを使用して、ネットワークを **up** としてマークしてください。) インターフェースが複数存在する場合、**routed** デーモンは、ローカル・ホストがネットワーク間でパケットを送信するものと見なします。インターフェースがサポートしていれば、**routed** デーモンは、ブロードキャスト・メッセージを使用して、各インターフェースで RIP 要求パケットを送信します。

次に、**routed** デーモンは、他のホストからの RIP 経路指定要求と応答パケットが送信されてくるのを待ちます。**routed** デーモンが RIP 情報を他のホストに提供する場合、直接接続されたホストとネットワークのすべてに対して 30 秒ごとに RIP 更新パケットを送信します (経路指定テーブルのコピーを含む)。

**routed** デーモンは、ルーティング情報プロトコル (RIP) 要求パケットを受信して RIP ルーティング情報を供給するときは、応答パケットのフォーマットで応答を作成します。応答パケットは、カーネル・ルーティング・テーブルに維持された情報を基本にしており、既知の経路のリストを含んでいます。各経路にはホップ・カウント・メトリックのマークが付けられていますが、このホップ・カウント・メトリックはソース・ネットワークと宛先ネットワーク間のゲートウェイ・ホップの数です。各経路のメトリックは、送信側のホストに関連します。16 以上のメトリックは無限度または到達不能と見なされます。

**routed** デーモンは、デーモン自体によるアクションに関する情報を *LogFile* に書き込みます。このログ・ファイルには、経路指定テーブルに対して行われた変更に関する情報が入っています。また、最近送受信した経路変更メッセージのヒストリーも入っています。

## 経路指定テーブルの更新

**routed** デーモンは、次の条件のうち 1 つでも成り立つと、他のホストからの RIP 応答や更新パケットに入っている情報を使用して、経路指定テーブルを更新します。

- 宛先ネットワークまたはホストの経路指定テーブル・エントリーが存在せず、経路指定に関連付けられたメトリックが限定されている (つまり、メトリックが 16 未満) 場合。
- パケットの送信元ホストが既存の経路指定テーブル・エントリーのルーターである場合。つまり、宛先へのパケットの経路が経由する同一のインターネットワーク・ルーターから更新済み情報が受信される場合。

- 経路指定テーブルの既存のエントリーがこの 90 秒間更新されておらず、経路指定が少なくとも現行の経路指定として有効な場合。
- 新しい経路指定が、経路指定テーブルに現在保管されている同一の宛先への経路より短い場合。**routed** デーモンは、新しいメトリックを経路指定テーブルに格納されているメトリックと比較して、経路の長さを相対的に判別します。

**routed** デーモンは、内部経路指定テーブルの更新時に、すべての直接接続ホストとネットワークに送信する RIP 更新パケットを生成します。**routed** デーモンは、カーネル経路指定テーブルを更新する前に一時的に停止して、不安定な条件を安定させます。

**routed** デーモンは、着信 RIP パケットを処理するだけでなく、定期的な内部経路指定テーブルの検査も行います。3 分間かけても更新が完了しなかったエントリーのメトリックは無限大に設定され、削除対象としてマークされます。無効な経路に関する情報をネットワーク上で配布できるように、削除は 60 秒間遅れて行われます。RIP ルーターとして動作するホストは、すべての直接接続ホストとネットワークにその経路指定テーブルを 30 秒ごとに提供します。

## ゲートウェイの使用法

**routed** デーモンは、直接接続のホストとネットワークへの経路を管理する機能の他に、遠隔ゲートウェイと外部ゲートウェイも使用します。これらのゲートウェイは RIP 照会では識別することができないため、**routed** デーモンは、外部ゲートウェイと遠隔ゲートウェイに関する情報を得るために **/etc/gateways** ファイルを読み取ります。

**/etc/gateways** ファイルには、ホストとネットワークへの遠隔ゲートウェイと外部ゲートウェイ経由の経路に関する情報が入っていて、この情報は RIP を通して伝達されます。このような経路は、特定の宛先への静的経路か、あるいは宛先への固定経路が未定義の場合に使用するデフォルト経路として使用できます。

**/etc/gateways** ファイルのフォーマットは、以下のようになります。

```
{ net | host } name1 gateway name2 metric{ passive | active | external }
```

**/etc/gateways** ファイルで指定されたゲートウェイが RIP ルーティング情報を提供する場合、このゲートウェイはアクティブであるとマークされます。アクティブなゲートウェイは、ネットワーク・インターフェースと同様に処理されます。つまり、RIP ルーティング情報はアクティブなゲートウェイに配布されます。RIP ルーティング情報が一定期間ゲートウェイから受信されない場合は、**routed** デーモンは経路指定テーブルから関連する経路指定を削除します。

RIP ルーティング情報を交換しないゲートウェイは、受動であるとマークされます。受動ゲートウェイは経路指定テーブルに固有に維持されています。受動ゲートウェイに関する情報は、送信される RIP ルーティング情報に含まれていません。

外部ゲートウェイが識別されて、別の経路指定プロセスが同様の経路指定を設置し、**routed** デーモンはこの宛先への代替経路指定を設置してはならないことが **routed** デーモンに通知されます。外部ゲートウェイは経路指定テーブルに維持されておらず、これらのゲートウェイに関する情報は送信される RIP ルーティング情報には含まれません。

注: 外部ゲートウェイを経由する経路は、必ずネットワークへの経路でなければなりません。

**routed** デーモンは、複数の異なるネットワークに経路を指定する場合にネーム・レゾリューションも行うことができます。例えば、以下に示すコマンドは、host1 というゲートウェイを経由する netname というネットワークへの経路を追加します。host1 ゲートウェイは 1 ホップ・カウント先です。

```
route add net netname host1 1
```



ネットワーク名のレゾリューションを行うために、**routed** デーモンは **/etc/networks** ファイルを使用してネットワーク・アドレスとそれに対応する名前の情報を獲得します。ホスト名のレゾリューションを行うために、**routed** デーモンは経路指定が完了する前に追加のステップを実行する必要があります。まず、デーモンは **/etc/resolv.conf** ファイルの存在をチェックします。このファイルは、ホストがドメイン・ネーム・サーバーの下で稼働しているかどうかを示し、稼働していれば **named** デーモンを稼働しているホスト・コンピュータの IP アドレスを表示します。

**/etc/resolv.conf** ファイルが存在しない場合、**routed** デーモンは、**/etc/hosts** ファイルを使用して経路指定先のホストを探します。

**routed** デーモンは、システム・リソース・コントローラー (SRC) または System Management Interface Tool (SMIT) を使用して制御しなければなりません。コマンド行に **routed** デーモンを入力することはお勧めできません。

## システム・リソース・コントローラーを使用した **routed** デーモンの操作

**routed** デーモンは、システム・リソース・コントローラー (SRC) によって制御されるサブシステムです。**routed** デーモンは SRC **tcpip** システム・グループのメンバーです。このデーモンはデフォルトでは使用不可で、以下の SRC コマンドにより操作できます。

|                  |                                              |
|------------------|----------------------------------------------|
| <b>startsrc</b>  | サブシステム、サブシステム・グループ、あるいはサブサーバーを開始します。         |
| <b>stopsrc</b>   | サブシステム、サブシステム・グループ、あるいはサブサーバーを停止します。         |
| <b>traceson</b>  | サブシステム、サブシステム・グループ、あるいはサブサーバーのトレースを使用可能にします。 |
| <b>tracesoff</b> | サブシステム、サブシステム・グループ、あるいはサブサーバーのトレースを使用不可にします。 |
| <b>lssrc</b>     | サブシステム、サブシステム・グループ、あるいはサブサーバーの状況を取得します。      |

## シグナル

以下のシグナルは、**kill** コマンドを使用して **routed** 処理に送信されると、指定された効果が得られます。

|                                     |                                                                                                                                                                                                                                                                            |
|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SIGINT</b>                       | <b>routed</b> デーモンを再始動して、経路指定テーブルをフラッシュします。                                                                                                                                                                                                                                |
| <b>SIGHUP, SIGTERM, または SIGQUIT</b> | ホップ・カウントを無限大に設定して RIP パケットをブロードキャストします。ルーターとして作動するローカル・ホストを使用不可にします。2 番目の <b>SIGHUP</b> 、 <b>SIGTERM</b> 、 <b>SIGQUIT</b> のいずれかのシグナルが送信されると、 <b>routed</b> デーモンは終了します。                                                                                                     |
| <b>SIGUSR1</b>                      | パケット・トレースをオンにするか、既にパケット・トレースがオンの場合には、トレース・レベルを 1 段階上げます。また、パケット・トレースが既にオンになっている場合は、トレースのレベルを 1 レベル上げます。第 1 レベルでは、トランザクションだけをトレースします。第 2 レベルでは、トランザクションとパケットをトレースします。第 3 レベルでは、パケット・ヒストリーをトレースし、パケットの変更を報告します。第 4 レベルでは、パケットの内容をトレースします。このコマンドは、この 4 つのレベルでトレースのレベルを増加させます。 |
| <b>SIGUSR2</b>                      | パケット・トレースをオフにします。                                                                                                                                                                                                                                                          |

## フラグ

**-d** 受信された正しくないパケットなどの追加のデバッグ情報がログに記録されるようにします。



- g** 経路指定デーモンをゲートウェイ・ホスト上で稼働します。**-g** フラグはインターネットワーク・ルーター上で使用され、その結果、デフォルトの宛先への経路を提供します。
- q** インターネットワーク・ルーターとして機能しているかどうかに関係なく、**routed** デーモンがルーティング情報を提供しないようにします。**-q** フラグは「抑制 (quiet)」を示します。**-q** フラグと **-s** を一緒に使用しないでください。
- s** インターネットワーク・ルーターとして機能しているかどうかに関係なく、ルーティング情報を提供します。**-s** フラグは「提供 (supply)」を意味します。**-q** フラグと **-s** を一緒に使用しないでください。
- t** 送受信したパケットをすべて標準出力または *LogFile* パラメーターで指定したファイルへ書き込みます。**routed** デーモンは、このデーモンを始動した制御端末の制御を受け続けます。したがって、**routed** プロセスは制御端末のキーボードからの割り込みによって停止します。

## 例

1. **routed** デーモンを手動で開始するには、次のように入力します。

```
startsrc -s routed -a "-s"
```

**注:** **routed** デーモンは、それぞれのシステムの開始時にデフォルトでは開始されません。**routed** デーモンを開始する場合は、**rc.tcpip** ファイル・フォーマットとシステム・リソース・コントローラー (SRC) のコマンドを使用してください。また、System Management Interface Tool (SMIT) を使用して **routed** デーモンを開始することもできます。

**-s** フラグを指定すると、**routed** デーモンがインターネットワーク・ルーターであるかどうかに関係なく、**routed** デーモンはルーティング情報を戻します。

2. **routed** デーモンを停止するには、次のように入力します。

```
stopsrc -s routed
```

3. **routed** デーモンからの簡略状況レポートを得るには、次のように入力します。

```
lssrc -s routed
```

このコマンドにより、デーモン名、デーモンのプロセス ID、デーモンの状態 (アクティブか、非アクティブか) が戻されます。

4. **routed** デーモンをトレース可能にするには、次のように入力します。

```
traceson -s routed
```

このコマンドは、ソケット・レベルのデバッグを使用可能にします。このコマンド例の出力結果を見る場合は、**trpt** コマンドを使用してください。

5. すべてのパケットのルーティング情報をトレースして一時ファイル *logfile* に記録するには、次のように入力します。

```
startsrc -s routed -a "-t /tmp/logfile"
```

## 関連情報

**route** コマンド。

**gated** デーモン。

**/etc/gateways** ファイル・フォーマット、**/etc/networks** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ の TCP/IP の経路指定、TCP/IP デーモン、TCP/IP のプロトコルのセクション。

---

## rpc.nisd デーモン

### 目的

NIS+ サービスをインプリメントします。

### 構文

```
/usr/sbin/rpc.nisd [ -A ] [ -C ] [ -D ] [ -F ] [ -h ] [ -v ] [ -Y ] [ -c Seconds ] [ -d Dictionary ] [ -L Load ] [ -S Level ]
```

### 説明

**rpc.nisd** デーモンは、NIS+ サービスをインプリメントするリモート・プロシージャー・コール・サービスです。このデーモンは、NIS+ ネーム・スペースの一部分にサービスを提供する全サーバーで実行してなければなりません。**rpc.nisd** は、通常システム始動スクリプトから始動されます。

### フラグ

- A** **rpc.nisd** デーモンを認証詳細モードで設定します。このデーモンは、すべての認証関連アクティビティを、**LOG\_INFO** の優先順位で **syslogd** にログに記録します。
- C** **/dev/console** で診断チャンネルを開きます。
- D** **rpc.nisd** デーモンをデバッグ・モード (fork しない) で設定します。
- F** サーバーが、その始動時にデータベースのチェックポイント指定を行うように強制します。強制されたチェックポイントは、サーバーのディスク・スペースが少なくなったときに必要になります。**-F** フラグは、すべてのレプリカに伝搬されたトランザクション・ログから更新を除去します。
- h** オプションのリストを印刷します。
- v** **rpc.nisd** デーモンを詳細モードで設定します。**-v** フラグを指定すると、**rpc.nisd** デーモンは、その操作の実行中のナレーションを **syslog** デーモンに送信します (**LOG\_INFO** 優先順位での **syslog** を参照)。このフラグは、NIS+ サービスに関する問題のデバッグに最も有効です (**-A** フラグも参照)。
- Y** サーバーを NIS (YP) 互換モードで設定します。このモードで作動しているとき、NIS+ サーバーは、バージョン 2 プロトコルを使用して、NIS バージョン 2 要求に応答します。YP プロトコルは認証されていないため、どこに対しても読み取りアクセスのないエントリだけがバージョン 2 プロトコルによって表示できます。バージョン 2 プロトコルは、このモードでは標準バージョン 2 マップしかサポートしません (**-B** フラグを参照)。
- c Seconds** サーバーのレプリカに対する更新の実行と実行の間の秒数を設定します。デフォルトは 120 秒 (2 分) です。
- d Dictionary** NIS+ データベースの代替辞書を指定します。**-d** フラグの基本的な用途はテストを行うことです。文字列は解釈されず、そのまま **db\_initialize** 機能に渡されます。
- L Load** サーバーが作成できる子プロセスの最大数を指定します。**Load** の値は、コールバック機能が正しく働くには少なくとも 1 でなければなりません。デフォルトは 128 です。

-S Level

**rpc.nisd** デーモンの許可セキュリティ・レベルを設定します。Level パラメーターの値は 0 から 2 の間でなければなりません。デフォルトは 2 です。次の値は、これらのセキュリティ・レベルを示しています。

- 0 セキュリティ・レベル 0 では、**rpc.nisd** デーモンは、アクセス制御を行使しません。どのクライアントも、任意の操作 (更新や削除など) を行うことができます。0 のセキュリティ・レベルは、NIS+ ネーム・スペースのテストと初期セットアップに使用されます。
- 1 セキュリティ・レベル 1 では、**rpc.nisd** デーモンは、クライアントを認証し、権限を与えるために、**AUTH\_SYS** と **AUTH\_DES** の両方の証明書を受け入れて、NIS+ 操作を行います。レベル 1 は、**AUTH\_SYS** 証明書が簡単に偽造できるため、操作は安全なモードではありません。このセキュリティ・レベルは、不明ユーザーがアクセスする可能性のあるネットワークでは使用しないようにしてください。
- 2 セキュリティ・レベル 2 では、**rpc.nisd** デーモンは、認証と許可用に **AUTH\_DES** 証明書だけを受け入れます。2 は NIS+ サービスが提供する最上位のセキュリティ・レベルで、デフォルトでもあります。

## 環境

**NETPATH** NIS+ が使用できるトランスポートを制限します。

## 例

1. NIS+ サービスをセットアップするには、次のように入力します。

```
rpc.nisd
```

2. NIS+ サービスを、DNS 転送を使用して YP 互換モードで設定するには、次のように入力します。

```
rpc.nisd -YB
```

## ファイル

**/var/nis/parent.object**

ルート・サーバーより上位のネーム・スペースを記述する、XDR エンコード NIS+ オブジェクトが入っています。この親ネーム・スペースは、別の NIS+ ネーム・スペースの場合も、ドメイン・ネーム・サーバーによってサービスを提供されるネーム・スペースなどの外部ネーム・スペースの場合もあります。 **/var/nis/parent.object** は、ルート・ドメイン・ネームスペースにサービスを提供するサーバー上にしかありません。

**/var/nis/root.object**

ネーム・スペースのルートを記述する XDR エンコード NIS+ オブジェクトが入っています。 **/var/nis/root.object** ファイルは、ネーム・スペースのルートにサービスを提供するサーバー上にしかありません。

**/etc/init.d/rpc**

NIS+ の初期化スクリプトが入っています。

## 関連情報

**nis\_cachemgr** デーモン、 **nisinit** コマンド、 **nissetup** コマンド、 および **rpc.nispasswd** デーモン。

---

## rpc.nispasswd デーモン

### 目的

NIS+ パスワード更新デーモン。

### 構文

```
/usr/sbin/rpc.nispasswd [ [ -a Attempts ] [ -c Minutes ] [ -D ] [ -g ] [ -v ]
```

### 説明

**rpc.nispasswd** デーモンは、**nispasswd** および **yppasswd** からパスワード更新要求へのサービスを行う ONC+ RPC サービスです。このデーモンは、NIS+ **passwd** テーブルのパスワード・エントリーを更新します。

**rpc.nispasswd** デーモンは、通常、NIS+ サーバーの **rpc.nisd** が始動された後に、システム始動スクリプトから始動されます。**rpc.nispasswd** は、それが 1 つまたは複数の NIS+ ディレクトリーのマスター・サーバーであるマシンで実行しているかどうかを判別します。ホストがマスター・サーバーでないことを確認すると、このデーモンは速やかに終了します。また、**rpc.nisd** が NIS(YP) 互換モードで実行しているかどうかを判別します (**-Y** フラグで、NIS(YP) クライアントの場合 **yppasswd** としても登録します)。

**rpc.nispasswd** デーモンは、失敗したパスワード更新の試みをすべて SYSLOG (システム・ログ) に記録します。これにより、管理者担当が、だれかがパスワードを「壊そうとした」かどうかを判別できます。

**rpc.nispasswd** はスーパーユーザーが実行させる必要があります。

### フラグ

|                    |                                                                                                                                                           |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-a Attempts</b> | パスワード更新要求セッション内でコール元を認証する場合に認められる、最大試行回数を設定します。失敗した試みは <b>syslogd</b> で、その要求はこのデーモンによってキャッシュされます。許容最大数に達すると、このデーモンは、クライアントとの接続を切断します。デフォルト値は 3 に設定されています。 |
| <b>-c Minutes</b>  | このデーモンが失敗したパスワード更新要求をキャッシュしなければならない分数を設定します。これは、このデーモンが同じユーザーについて別のパスワード更新要求を受信して、コール元の認証が失敗した場合に、このデーモンが単に応答しないままにいる時間です。デフォルト値は 30 分に設定されています。          |
| <b>-D</b>          | デバッグ・モードで実行します。                                                                                                                                           |
| <b>-g</b>          | DES 証明書を生成します。デフォルトでは、DES 証明書は、証明書を持たないユーザーの場合には生成されません。このフラグを指定すると、ユーザーが証明書を持たない場合、そのユーザー用に証明書が生成され、NIS+ cred テーブル内にそれが保管されます。                           |
| <b>-v</b>          | 詳細モードを設定します。このフラグを指定すると、このデーモンは、それが現在行っている内容についての実行中のナレーションを <b>syslog</b> デーモンに送信します。このフラグは、問題をデバッグする場合に役に立ちます。                                          |

### 終了状況

|   |             |
|---|-------------|
| 0 | 正常終了        |
| 1 | エラーが発生しました。 |

## ファイル

/etc/init.d/rpc

NIS+ の初期化スクリプト

## 関連情報

**passwd** コマンド、 **yppasswd** コマンド。

**rpc.nisd** デーモン、 **syslogd** デーモン。

---

## rpc.pcnfsd デーモン

### 目的

PC-NFS (パーソナル・コンピューター・ネットワーク・ファイルシステム) クライアントからのサービス要求を処理します。

### 構文

/usr/sbin/rpc.pcnfsd

### 説明

**rpc.pcnfsd** デーモンは、PC-NFS クライアントからのリモート・コンピューターの認証サービス要求を処理します。このようなサービスには、マウントや印刷スプーリングのための認証も含まれています。PC-NFS プログラムにより、DOS を実行しているパーソナル・コンピューターと NFS を実行しているコンピューターをネットワークで相互接続することができます。 **rpc.pcnfsd** デーモンは、バージョン 1 と 2 の **pcnfsd** プロトコルをサポートしています。

PC-NFS クライアントが要求を実行すると、**inetd** デーモンは **rpc.pcnfsd** デーモンを開始します (**inetd.conf** ファイルに該当するエントリが入っている場合)。**rpc.pcnfsd** デーモンは、**/etc/pcnfsd.conf** 構成ファイルが存在する場合はこのファイルを読み取り、プログラム番号 150001 あてに出された RPC 要求を維持します。**rpc.pcnfsd** デーモンが開始されると、印刷要求はすべてデフォルトの印刷スプール・ディレクトリー (**/var/spool/pcnfs**) に入ります。

### 認証

**rpc.pcnfsd** デーモンは、**PCNFSD\_AUTH** 要求または **PCNFSD2\_AUTH** 要求を受信すると、まずユーザーの名前とパスワードの妥当性を確認し、次に対応するユーザー ID (UID)、グループ ID (GIDS)、ホーム・ディレクトリー、**umask** 指定を戻して、ユーザーのログインを行います。ログインの記録は、**/var/adm/wtmp** ファイルに追加されます。ログイン・レコード機能を使用不可にするには、**/etc/pcnfsd.conf** ファイルに次の行を追加します。

```
wtmp off
```

### 印刷

**rpc.pcnfsd** デーモンはネットワーク・ファイルシステム (NFS) を使用して印刷データをクライアント/サーバー間で転送します。クライアント・システムはまず、**PCNFSD\_PR\_INIT** 要求または **PCNFSD2\_PR\_INIT** 要求を出します。次に、サーバーがクライアントが使用するスプール・ディレクトリーのパスを戻します。

注: スプール・ディレクトリーは、必ず NFS によって **exportfs** コマンドと **/etc/exports** ファイルを使用してエクスポートする必要があります。

**rpc.pcnfsd** デーモンは各クライアントのサブディレクトリーを作成します。デフォルトの親ディレクトリーは **/var/spool/pcnfs** で、サブディレクトリーはクライアント・システムのホスト名です。別の親ディレクトリーを使用する場合は、**/etc/pcnfsd.conf** ファイルに次の行を追加します。

spooldir Pathname

スプール・ディレクトリーがマウントされ、このディレクトリー内のファイルに印刷データが転送されると、クライアントが **PCNFSD\_PR\_START** 要求または **PCNFSD2\_PR\_START** 要求を出します。

**rpc.pcnfsd** デーモンは **enq** コマンドを作成して、この要求 (およびほとんどの印刷関連要求) を処理します。このデーモンは、パーソナル・コンピューターのユーザーの ID を承認して印刷要求コマンドを実行します。コマンドの作成と実行にはユーザー ID 特権も使用されるため、**rpc.pcnfsd** デーモンはルート・プロセスとして実行する必要があります。

すべてのクライアントからの印刷要求には、使用するプリンターの名前が含まれています。プリンター名は、**/etc/qconfig** ファイルに入っているキュー定義とデバイス定義によって表されます。また、**rpc.pcnfsd** デーモンは、**rpc.pcnfsd** クライアントだけが認識する PC-NFS 仮想プリンターを定義するメソッドを提供します。各 PC-NFS 仮想プリンターは次のような行で **/etc/pcnfsd.conf** ファイルに定義されます。

printer Name AliasFor Command

上記のフォーマットでは、Name に定義するプリンターの名前を指定し、AliasFor に実際に処理を行う既存のプリンターを指定します。例えば、Name のキューを表示するという要求は、プリンター AliasFor のキュー・コマンドに変換されます。存在しないプリンターを使用してプリンター Name を定義するには、AliasFor パラメーターに - (負符号) を 1 つ指定します。Command パラメーターには、プリンター Name でファイルを印刷するときの実行するコマンドを指定します。このコマンドは Bourne シェルが **-c** オプションを使用して実行します。複合操作の場合、Command パラメーターは実行可能なシェル・スクリプトに置換されます。

Command パラメーターで使用できるトークンと置換値のリストを以下に示します。

| トークン          | 置換値                                         |
|---------------|---------------------------------------------|
| <b>\$FILE</b> | 印刷データ・ファイルの絶対パス名。コマンドの実行後に、ファイルのリンクは解除されます。 |
| <b>\$USER</b> | クライアントへログインしたユーザーの名前。                       |
| <b>\$HOST</b> | クライアント・システムのホスト名。                           |

## 例

次の例では、**/etc/pcnfsd.conf** ファイルが 1 行目で仮想プリンターを構成し、2 行目でテスト用の null デバイスを構成します。

```
printer rotated lw /bin/enscript -2r $FILE
printer test - /usr/bin/cp $FILE /usr/tmp/$HOST-$USER
```

1 行目は、クライアント・システムがプリンター rotated でジョブを印刷する場合は、enscript ユーティリティーが呼び出されて \$FILE ファイルのプリプロセスを行うことを示します。オプション -2r によって、ファイルは 2 桁の循環フォーマットでデフォルトの PostScript プリンターから印刷されます。クライアントが rotated プリンターの印刷キューのリストを要求すると、**rpc.pcnfsd** デーモンはこの要求をプリンター lw の同様のリストの要求に変換します。



2 行目では、プリンターのテストが確立されます。test プリンターに送られるファイルが **/usr/tmp** ディレクトリーにコピーされます。キューのリスト化、状況検査、または同様のプリンター操作の実行といったことを test プリンターへ要求しても、*AliasFor* パラメーターに - (負符号) が指定されているため、リジェクトされます。

## ファイル

|                         |                                                      |
|-------------------------|------------------------------------------------------|
| <b>/etc/inetd.conf</b>  | RPC デーモンおよび他の TCP/IP デーモンを開始する TCP/IP 構成ファイルが入っています。 |
| <b>/etc/pcnfsd.conf</b> | <b>rpc.pcnfsd</b> デーモン構成ファイルが入っています。                 |
| <b>/var/spool/pcnfs</b> | デフォルトの印刷スプール・ディレクトリーが入っています。                         |

## 関連情報

**enq** コマンド、**last** コマンド。

**inetd** デーモン。

**umask** サブルーチン。

ネットワークおよびコミュニケーションのマネージ のネットワーク・ファイルシステム (NFS) の概要のセクション。

プリンターおよび印刷 の印刷の管理 のセクション。

ネットワークおよびコミュニケーションのマネージ の NFS コマンドのリストのセクション。

---

## rpcgen コマンド

### 目的

RPC プロトコルをインプリメントするための C コードを生成します。

### 構文

#### 1 つのファイルに対して 4 タイプの出力ファイルを生成する

```
/usr/bin/rpcgen InputFile
```

#### 1 つのファイルに対して特定の出力ファイルを生成する

```
rpcgen { -c | -h | -l | -m } [ -o OutputFile ] [ InputFile ]
```

#### TCP または UDP に対してサーバー側のファイルを生成する

```
rpcgen { -s Transport ... } [ -o OutputFile ] [ InputFile ]
```

### 説明

**rpcgen** コマンドは、リモート・プロシージャ・コール (RPC) プロトコルをインプリメントするための C コードを生成します。RPC **rpcgen** コマンドへの入力、RPC 言語として知られている C 言語に似た言語です。



最初の構文構造は、**rpcgen** コマンドに使用される最も一般的なフォーマットで、1つの入力ファイルから4つの出力ファイルを生成します。例えば、*InputFile* パラメーターの名前が **proto.x** であれば、**rpcgen** コマンドは下記のようなファイルを生成します。

|                     |            |
|---------------------|------------|
| <b>proto.h</b>      | ヘッダー・ファイル  |
| <b>proto_xdr.c</b>  | XDR ルーチン   |
| <b>proto_svc.c</b>  | サーバー側スタブ   |
| <b>proto_clnt.c</b> | クライアント側スタブ |

他の構文構造は、4つのすべての出力ファイルではなく、特定の出力ファイルを生成したいときに使用します。

**cpp** コマンド (C プリプロセッサ) は、実際に **rpcgen** コマンドによって解釈される前にすべての入力ファイル上で実行されます。したがって、すべての **cpp** 指示は **rpcgen** 入力ファイル内で正当です。出力ファイルの各タイプに対して、**rpcgen** コマンドは、**rpcgen** プログラマーによって使用される特別な **cpp** 記号を次のように定義します。

|                 |                               |
|-----------------|-------------------------------|
| <b>RPC_HDR</b>  | ヘッダー・ファイル内にコンパイルされる際に定義されます。  |
| <b>RPC_XDR</b>  | XDR ルーチン内にコンパイルされる際に定義されます。   |
| <b>RPC_SVC</b>  | サーバー側スタブ内にコンパイルされる際に定義されます。   |
| <b>RPC_CLNT</b> | クライアント側スタブ内にコンパイルされる際に定義されます。 |

さらに、**rpcgen** コマンドはそれ自体のプリプロセスをいくつか行います。**%** (パーセント記号) で始まる行は、直接出力ファイルに送られ、**rpcgen** コマンドでは解釈されません。

ユーザーが XDR ルーチンを作成するには、データ・タイプを定義しないでおきます。すべての定義されていないデータ・タイプに対しては、**rpcgen** コマンドは、未定義のタイプ名に **xdr\_** を事前に保留して、ルーチンが存在するものと見なします。

#### 注:

1. ネスト機能はサポートされていません。他の構造内でその名前を使用して、全体に渡って、構造をトップレベルで宣言すれば、ネストと同じ効果が得られます。
2. 有効範囲が明確に指定されていないので、プログラム定義を使用する際にネーム・クラッシュが発生することがあります。一般にこのような事態は、プログラム、バージョン、プロシージャおよびタイプに固有の名前を付けて避けることができます。
3. TIRPC インターフェースに対するプログラムを作成し、マルチスレッド RPC アプリケーションを使用できるようにするには、**tirpcgen** コマンドを使用します。また、プリプロセッサ変数 **\_AIX\_TIRPC** を MAKE ファイルおよび **libtli.a (-ltli)** 仕様に定義する必要があります。**tirpcgen** は、新規 **rpcgen** コマンドの一時的な名前です。将来のバージョンのオペレーティング・システムで **rpcgen** に取って代わるものです。

## フラグ

|                      |                                                                                            |
|----------------------|--------------------------------------------------------------------------------------------|
| <b>-c</b>            | XDR ルーチンにコンパイルします。                                                                         |
| <b>-h</b>            | C データ定義 (ヘッダー・ファイル) 内にコンパイルします。                                                            |
| <b>-l</b>            | クライアント側スタブ内にコンパイルします。                                                                      |
| <b>-m</b>            | サーバー側スタブ内にコンパイルしますが、メイン・ルーチンは生成しません。このオプションは、コールバック・ルーチンを実行したり、初期化を行うメイン・ルーチンを書き込む際に役立ちます。 |
| <b>-o OutputFile</b> | 出力ファイルの名前を指定します。何も指定されない場合、標準出力が使用されます。                                                    |

**-s** *Transport*

与えられたトランスポートを使用して、サーバー側スタブ内にコンパイルします。サポートされているトランスポートは `udp` と `tcp` です。このフラグは、複数のトランスポートにサービスを提供するサーバーをコンパイルするために、2 回以上実行できます。

## 関連情報

**cpp** コマンド。

ネットワークおよびコミュニケーションのマネージ のネットワーク・ファイルシステム (NFS) の概要のセクション。

「*AIX 5L Version 5.3 Communications Programming Concepts*」の Remote Procedure Call (RPC) Overview for Programming のセクション。

ネットワークおよびコミュニケーションのマネージ の NFS コマンドのリストのセクション。

---

## rpcinfo コマンド

### 目的

リモート・プロシージャ・コール (RPC) サーバーの状況を報告します。

### 構文

#### 統計情報のリストを表示する

```
/usr/bin/rpcinfo [ -m | -s ] [ Host ]
```

#### 登録済みの RPC プログラムのリストを表示する

```
/usr/bin/rpcinfo -p [ Host ]
```

#### トランスポートを報告する

```
/usr/bin/rpcinfo -T transport Host Prognum [ Versnum ]
```

#### エントリー・リストを表示する

```
/usr/bin/rpcinfo -l [ -T transport ] Host Prognum Versnum
```

#### UDP を使用してプログラム状況を報告する

```
/usr/bin/rpcinfo [-n PortNum] -u Host Prognum [ Versnum ]
```

#### TCP を使用してプログラム状況を報告する

```
/usr/bin/rpcinfo [ -n PortNum ] -t Host Prognum[ Versnum ]
```

#### プログラム状況を報告する

```
/usr/bin/rpcinfo -a ServAddress -T transport Host Prognum [ Versnum ]
```

#### 指定されたプログラム・バージョンを実行しているすべてのホストを表示する

```
/usr/bin/rpcinfo [ -b ] [ -T transport ] Prognum Versnum
```

## サービスの登録を削除する

`/usr/bin/rpcinfo [ -d ] [ -T transport ] Prognum Versnum`

### 説明

**rpcinfo** コマンドは RPC サーバーへの RPC コールを行い、サーバーの状況を報告します。例えば、サーバーが使用可能な状態で待機しているか、または使用不可かどうかを報告します。

プログラム・パラメーターには名前または数値を指定できます。バージョンを指定すると、**rpcinfo** コマンドは、指定されたプログラムのそのバージョンをコールしようとします。バージョンを指定しなければ、**rpcinfo** コマンドは、バージョン 0 (ゼロ) をコールして、ユーザーが指定したプログラムのすべての登録済みのバージョン番号を探し出してから、登録済みのバージョンをそれぞれ呼び出そうとします (バージョン 0 は存在しないものと見なされています。存在する場合は、**rpcinfo** コマンドが、バージョン 0 の代わりに極端に高いバージョン番号を呼び出して、この情報を得ようとしています)。

### フラグ

- a** ホストの完全な IP アドレスとポート番号を指定します。
- b** 指定したプログラム番号およびバージョン番号のプロシーチャー 0 への RPC ブロードキャストを行い、応答するホストをすべて報告します。 トランスポート を指定すると、その要求が指定した トランスポート 上でのみブロードキャストされます。ブロードキャストがどの トランスポート によってもサポートされない場合は、エラー・メッセージが出力されます。ブロードキャスト (**-b** フラグ) の使用は、他のシステムに逆効果を与えることがあるので制限されます。
- d** 指定したプログラム番号およびバージョン番号についての RPC サービスに関する登録を削除します。トランスポートを使用する場合は、そのトランスポート上でのみ行われるサービスを抹消します。それ以外の場合は、サービスを登録したすべてのトランスポートで行われるサービスを抹消してください。このオプションを実行できるのは、root ユーザーだけです。
- l** 指定したホストに、指定したプログラム番号およびバージョン番号を持つエントリーのリストを表示します。エントリーは、リモートの **portmap** デモンに接続するときを使用するのと同じプロトコル・ファミリーで、すべてのトランスポートに対して戻されます。このフラグは、AIX バージョン 4.2.1 以降のバージョンにのみ適用されます。
- m** 指定したホストに、ポート・マップ操作統計情報のテーブルを表示します。テーブルには、ポート・マップ (AIX バージョン 2、3、および 4) の各バージョンの統計情報、各プロシーチャーが要求され、正常にサービスを提供された回数、作成されたりリモート・コール要求の数とタイプ、およびハンドルされた RPC アドレス検索についての情報が含まれます。この情報は、ホストでの RPC アクティビティーをモニターするときに使用します。このフラグは、AIX バージョン 4.2.1 以降のバージョンにのみ適用されます。
- n Portnum** ポート・マップによって指定されたポート番号の代わりに、**-t** および **-u** オプションのポート番号である *Portnum* パラメーターを使用します。**-n** オプションを使用すると、サービスのアドレスを検索するためのリモート・ポート・マップへの呼び出しをする必要はありません。このオプションは、**-a** オプションによって取って代わられます。
- p** ポート・マップ・プロトコルのバージョン 2 を使用して、ホストで **portmap** サービスを照会して、登録されているすべての RPC プログラムのリストを表示します。ホストを指定しない場合は、ローカル・ホストがデフォルトになります。
- s** 登録されているすべての RPC プログラムの簡単なリストを、ホストに表示します。ホストを指定しない場合、デフォルトはローカル・ホストです。このフラグは、AIX バージョン 4.2.1 以降のバージョンにのみ適用されます。
- t** TCP を使用して、指定されたホスト上で、プログラム番号のプロシーチャー 0 に対する RPC 呼び出しを行い、応答が受信されたかどうかを報告します。3 番目の構文に示されるように **-T** オプションを使用すると、このオプションは廃止されます。
- T** サービスを要求するトランスポートを指定します。このフラグは、AIX バージョン 4.2.1 以降のバージョンにのみ適用されます。

- u** 指定されたホスト上で、UDP を使用してプログラム番号のプロシージャ 0 に対する RPC 呼び出しを行い、応答が受信されたかどうかを報告します。3 番目の構文に示されるように **-T** オプションを使用すると、このオプションは廃止されます。

## 例

1. ローカル・コンピューターに登録されているすべての RPC サービスを表示するには、下記のように入力します。

```
rpcinfo -p
```

2. 特定のコンピューターに登録されているすべての RPC サービスを表示するには、次のように入力します。

```
rpcinfo -p zelda
```

この例で、**rpcinfo** コマンドは **zelda** というコンピューターに登録されている RPC サービスをすべて表示します。

3. 特定のサーバーのあるバージョンを実行しているローカル・ネットワーク上のコンピューターをすべて表示するには、次のように入力します。

```
rpcinfo -b ypserv 2
```

この例では、**rpcinfo** コマンドは **ypserv** デーモンのバージョン 2 を実行しているすべてのコンピューターのリストを表示します。

4. サービスの登録を削除するには、次のように入力します。

```
rpcinfo -d sprayd 1
```

この例では、**rpcinfo** コマンドは **sprayd** デーモンのバージョン 1 を削除します。

5. IP アドレス 127.0.0.1、プログラム 100003、およびバージョン 3 が指定されたホストが TCP 上でポート 2049 を listen しているかどうかをチェックするには、次のように入力します。

```
rpcinfo -a 127.0.0.1.8.1 -T tcp 100003 3
```

## ファイル

### **/etc/rpc**

サーバー名およびそれに対応する RPC プログラム番号と別名のリストが入っています。

### **/etc/services**

インターネット・ネットワーク全体で使用できる各サービスのエントリーが入っています。

## 関連情報

**portmap** デーモン。

ネットワークおよびコミュニケーションのマネージ のネットワーク・ファイルシステム (NFS) の概要のセクション。

ネットワークおよびコミュニケーションのマネージ の NFS コマンドのリストのセクション。

---

## rrestore コマンド

### 目的

既にバックアップを取っているファイルシステムを、リモート・マシンのデバイスからローカル・マシンへコピーします。

### 構文

```
rrestore [ -bNumber ][ -h ][ -i ][ -m ][ -sNumber ][ -t ][ -v ][ -y ][ -x ][ -r ]  
[ -R ] -fMachine:Device [ FileSystem ... ][ File ... ]
```

### 説明

**rrestore** コマンドは、i ノードのバックアップを使用して、リモート・マシンのデバイスからローカル・マシン上のファイルシステムへバージョン 3 を復元します。**rrestore** コマンドは、リモート・マシン上にサーバーを作成し、これをバックアップ・メディアとして使用します。

**rrestore** コマンドは、ファイルシステムのバックアップが i ノードにより作成された場合に、作成されたバックアップ・フォーマットだけを受け入れます。

注: ユーザーは、このコマンドを実行するための root 権限を持っていない限りなりません。

### フラグ

- |                         |                                                                                                                                                      |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-bNumber</b>         | 1 回の入力操作で読み取るブロック数を指定します。このフラグを指定しない場合、 <b>rrestore</b> コマンドにより、選択した物理デバイスに適切なデフォルト値が選択されます。 <i>Number</i> 変数の値が大きいほど、テープ・デバイスからの物理転送量が多くなります。       |
| <b>-fMachine:Device</b> | リモート・マシンの入力デバイスを指定します。特定のデバイスから入力するには、 <i>Device</i> 変数にファイル名 ( <b>/dev/rmt0</b> ファイルなど) を指定してください。テープ・デバイスの使用については、 <b>rmt</b> スペシャル・ファイルを参照してください。 |
| <b>-h</b>               | <i>File</i> パラメーターで指定されたディレクトリーに入っているファイルではなく実ディレクトリーだけを復元します。このオプションは、 <b>-r</b> フラグまたは <b>-R</b> フラグが指定されている場合は無視されます。                             |

**-i**

対話モードを開始します。このフラグを使用すれば、*File* パラメーターで表されたディレクトリーから選択したファイルだけを復元することができます。**-i** フラグ用のサブコマンドは、下記のとおりです。

**ls** [*Directory*]

*Directory* パラメーターに指定されたディレクトリー名の後に / (スラッシュ) が付けられた場合には、そのディレクトリー内のディレクトリー名を表示し、ディレクトリー名の前に \* (アスタリスク) を付けて復元するファイルを表示します。**-v** フラグを使用すると、それぞれのファイルおよびディレクトリーの *i* ノード番号も表示されます。*Directory1* パラメーターを指定しないと、現行ディレクトリーが使用されます。

**cd** *Directory*

現行ディレクトリーを *Directory* パラメーターに変更します。

**pwd** 現行ディレクトリーの絶対パス名を表示します。

**add** [*File*]

復元する *File* パラメーターを指定します。*File* パラメーターがディレクトリーの場合には、そのディレクトリーとディレクトリー内の全ファイルが復元されます (ただし、**-h** フラグが使用されている場合は除きます)。復元されるファイルは、**ls** サブコマンドにより名前の前に \* (アスタリスク) を付けて表示されます。*File* パラメーターを指定しないと、現行ディレクトリーが使用されます。

**delete** [*File*]

復元時に無視すべき *File* パラメーターを指定します。*File* パラメーターがディレクトリーの場合には、そのディレクトリーとディレクトリー内の全ファイルが復元されません (ただし、**-h** フラグが使用されている場合は除きます)。*File* パラメーターを指定しないと、現行ディレクトリーが使用されます。

**extract** **ls** サブコマンドを使用して、名前の前に \* (アスタリスク) が付けられた場合に表示されるファイルをすべて復元します。

**setmodes**

バックアップ・メディアに入っている復元ファイルのオーナー、モード、回数  
の情報を使用せずに、これらを設定します。

**verbose**

**ls** サブコマンドで復元されるファイルすべての *i* ノード番号を表示します。ファイルが復元されると、各ファイルの情報も表示されます。**verbose** サブコマンドを再度呼び出すと、**verbose** がオフになります。

**help** サブコマンドの要約を表示します。

**quit** 要求されたファイルがすべて復元されていなくても、**rrestore** コマンドの実行を即座に終了します。

**-m**

パス名ではなく、*i* ノード番号を使用してファイルを復元します。



|                 |                                                                                                                                                                                                                                                                                                                                               |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-r</b>       | ファイルシステム全体を復元します。<br><b>重要:</b> よく注意してこの手順に従わないと、ファイルシステム全体を破壊してしまうことがあります。(レベル 0) のフル・バックアップを復元する場合には、 <b>mkfs</b> コマンドを実行して、復元を行う前に空のファイルシステムを作成してください。例えば、レベル 2 の増分バックアップを復元するには、 <b>mkfs</b> コマンドを実行して、該当のレベル 0 のバックアップを復元し、次にレベル 1 のバックアップを復元し、最後にレベル 2 のバックアップを復元してください。安全に対する予防措置として、それぞれのバックアップ・レベルを復元したら、 <b>fsck</b> コマンドを実行してください。 |
| <b>-R</b>       | <b>rrestore</b> コマンドでファイルシステム全体を復元する際に、バックアップ・メディアの複数のボリューム・セットの中から特定のボリュームを要求します。 <b>-R</b> フラグを使用すると、 <b>rrestore</b> コマンドを中断、再開することができます。                                                                                                                                                                                                  |
| <b>-sNumber</b> | 複数のバックアップ・メディアから復元するバックアップを指定します。番号は 1 から始まります。                                                                                                                                                                                                                                                                                               |
| <b>-t</b>       | バックアップを作成したファイルの目次を表示します。 <b>rrestore</b> コマンドは、ファイル名を表示します。この名前は、バックアップが作成されたファイルシステムのルート ( <i>/</i> ) ディレクトリーに対する相対名です。ただし、ルート ( <i>/</i> ) ディレクトリーのみは例外です。                                                                                                                                                                                 |
| <b>-v</b>       | 復元が進行するにつれて、その進行状況を報告します。                                                                                                                                                                                                                                                                                                                     |
| <b>-x</b>       | 個々に指定されたファイルを復元します。名前が指定されていない場合には、そのメディア上の全ファイルが復元されます。名前は、 <b>-t</b> フラグで表示された名前と同じフォーマットのものでなければなりません。                                                                                                                                                                                                                                     |
| <b>-y</b>       | <b>rrestore</b> コマンドの実行時にテープのエラーが発生しても、復元を中止するかどうかを尋ねないように指定します。 <b>rrestore</b> コマンドは、可能な限り不良ブロックをスキップしようとします。                                                                                                                                                                                                                               |
| <b>-?</b>       | 使用方法メッセージを表示します。                                                                                                                                                                                                                                                                                                                              |

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0 正常終了。
- >0 エラーが発生しました。

## 例

1. リモートのテープ・デバイス上のファイルをリストするには、次のように入力します。

```
rrestore -fmachine1:/dev/rmt0 -t
```

このコマンドにより、リモートの machine1 上の /dev/rmt0 デバイスの情報が読み取られます。ファイル名が表示されます。

2. ファイルを復元するには、次のように入力します。

```
rrestore -x -fmachine1:/dev/rmt0 /home/mike/file1
```

このコマンドにより、リモートの machine1 上の /dev/rmt0 デバイス上にあるバックアップ・メディアから /home/mike/file1 ファイルが取り出されます。

3. ディレクトリー内の全ファイルを復元するには、次のように入力します。

```
rrestore -fhost:/dev/rmt0 -x /home/mike
```



このコマンドにより、ディレクトリー /home/mike と、そのディレクトリー内の全ファイルが復元されます。

4. ディレクトリー内のファイルは復元せず、ディレクトリーだけを復元するには、次のように入力します。

```
rrestore -fhost:/dev/rmt0 -x -h /home/mike
```

5. 複数バックアップのメディア上の特定のバックアップから、ディレクトリー内の全ファイルを復元するには、次のように入力します。

```
rrestore -s3 -fhost:/dev/rmt0.1 -x /home/mike
```

このコマンドにより、バックアップ・メディア上の 3 番目のバックアップから、ディレクトリー /home/mike と、そのディレクトリー内の全ファイルが復元されます。

## ファイル

**/dev/rfd0** デフォルトの復元デバイスを指定します。  
**/usr/sbin/rrestore** **rrestore** コマンドが入っています。

## 関連情報

**backup** コマンド、**rdump** コマンド、**restore** コマンド、**mkfs** コマンド、**fsck** コマンド、**dd** コマンド。

**filesystems** ファイル、**backup** ファイル、**rmt** スペシャル・ファイル。

オペレーティング・システムおよびデバイスのマネージのバックアップ方式のセクション。

オペレーティング・システムおよびデバイスのマネージのファイルシステムのセクション。

オペレーティング・システムおよびデバイスのマネージのディレクトリーのセクション。

オペレーティング・システムおよびデバイスのマネージのファイルのセクション。

---

## Rsh コマンド

### 目的

制限付きの Bourne シェルを起動します。

### 構文

```
Rsh [-i] [ { + | - } { [ a ] [ e ] [ f ] [ h ] [ k ] [ n ] [ t timeout ] [ u ] [ v ] [ x ] } ] [-c String | -s | File [ Parameter ] ]
```

注: - (負符号) ではなく、+ (正符号) をフラグの前に付けると、フラグのセットが解除されます。

### 説明

**Rsh** コマンドは制限付きの Bourne シェルを起動し、より制御されたシェル環境を必要とするインストールで有効です。制限付きのシェルにより、限定された特権や機能のセットを持つユーザー環境を作成できます。

Bourne シェルの詳細は、オペレーティング・システムおよびデバイスのマネージの Bourne シェルを参照してください。

## フラグ

Bourne シェルはフラグがコマンド行で起動された時にのみ次のフラグを解釈します。

注: **-c** または **-s** フラグを指定しない限り、シェルは次のパラメーターをコマンド・ファイル (シェル・スクリプト) であると判断します。それ以外のフラグはすべてコマンド行からコマンド・ファイルへ引き渡されます。定位置パラメーターについては、オペレーティング・システムおよびデバイスのマネージの Bourne シェルにおける変数置換を参照してください。

- a** 割り当てられるものに対し、すべての変数をエクスポートすることを示します。コマンド名よりも先に割り当てがくる場合、エクスポート属性はそのコマンドの実行環境に対してのみ有効です (ただし、割り当てが特定の組み込みコマンドのうちの 1 つより先にくる場合を除きます)。この場合、エクスポート属性は組み込みコマンドが完了した後に持続します。割り当てがコマンド名よりも先でない場合、または割り当てが **getopts** または **read** コマンドの結果である場合、エクスポート属性は変数のセットが解除されるまで持続します。
- c String** *String* 変数から読み取ったコマンドを実行します。 *String* 変数の値から特定パラメーター 0 の値を設定し、残りの *Parameter* オペランドから順番に定位置パラメーター (**\$1**、**\$2** など) の値を設定します。このフラグを指定した場合、シェルは標準入力から追加のコマンドを読み取りません。
- e** コマンドに対して次の条件がすべて存在する場合、ただちに終了します。
  - 0 より大きい戻り値で終了する。
  - **while**, **until** または **if** コマンドの複合リストの部分でない。
  - AND または OR リストを使用してテストされていない。
  - (感嘆符) 予約語が前に付くパイプライン処理でない。
- f** ファイル名の置換を使用不可にする。
- h** ファンクションが定義にされた時に、ファンクション内で呼び出されたコマンドを検索し記憶します。(通常これらのコマンドはファンクションの実行時に探し出されます。 **hash** コマンドを参照してください。)
- i** 入出力がワークステーションからでない場合でも、シェルを対話式にします。この場合、シェルは **TERMINATE** シグナルを無視します。そのため **kill 0** コマンドは対話式シェルを停止せず、**INTERRUPT** シグナルをトラップするため **wait** コマンドの機能に割り込みを行うことができます。すべての場合において、シェルは **QUIT** シグナルを無視します。
- k** 前にあるコマンド名のみでなく、すべてのキーワード・パラメーターをコマンドの環境に配置します。
- n** コマンドを読み込みますが、実行しません。**-n** フラグはシェル・スクリプトの構文エラーを検査するのに使用します。対話式シェルはこのオプションを無視する場合があります。
- s** 標準入力からコマンドを読み込みます。指定された任意の他のパラメーターは、新しいシェルへ定位置パラメーターとして引き渡されます。シェルの出力は、組み込みコマンドの出力以外は標準エラーへ書き込まれます。
- t timeout** サーバーからの応答がない場合、*timeout* 秒後に存在します。
- u** 変数の置換を行う際に、セットを解除された変数をエラーとして取り扱い、ただちに終了します。対話式シェルは存在しません。
- v** 読み込まれたシェルの入力行を表示します。
- x** コマンドと引数を実行前に表示します。

注: - (負符号) でなく + (正符号) を使用するとフラグのセットが解除されます。**\$-** 特殊変数は現在のフラグのセットを含みます。

## ファイル

`/usr/bin/bsh` Bourne シェルへのパス名を指定します。  
`/usr/bin/Rsh` 制限付き Bourne シェル (Bourne のサブセット) へのパス名を指定します。  
`/tmp/sh*` シェルがオープンした時に作成された一時ファイルを含みます。

## 関連情報

`env` コマンド、`sh` コマンド、`bsh` コマンド。

`/etc/passwd` ファイル、`null` スペシャル・ファイル、`environment` ファイル。

`profile` ファイル・フォーマット。

オペレーティング・システムおよびデバイスのマネージの Bourne シェルのセクション。

オペレーティング・システムおよびデバイスのマネージの Bourne シェル組み込みコマンドのセクション。

オペレーティング・システムおよびデバイスのマネージの Bourne シェルにおける変数置換のセクション。

---

## rsh または remsh コマンド

### 目的

指定したコマンドをリモート・ホスト側で実行するか、またはリモート・ホストにログインします。

### 構文

```
{ rsh | remsh } RemoteHost [ -a ] [ -n ] [ -l User ] [ -f | -F ] [ -k realm ] [ -S ] [ -u ] [ Command ]
```

### 説明

`/usr/bin/rsh` コマンドは、*Command* パラメーターで指定したコマンドを、*RemoteHost* パラメーターで指定したリモート・ホスト側で実行します。*Command* パラメーターを指定しない場合には、`rsh` コマンドが *RemoteHost* パラメーターで指定したリモート・ホストにログインします。`rsh` コマンドは、ローカル・コマンド行からリモート・コマンドへ標準入力を送信し、リモート・コマンドから標準出力と標準エラーを受信します。

注: リモート・コマンドへの入力は必ずローカル・コマンド行で指定しなければならないため、`rsh` コマンドを使用してリモート・ホスト上で対話式コマンドを実行することはできません。リモート・ホスト上で対話式コマンドを実行する必要がある場合には、`rlogin` コマンドを使用するか、または *Command* パラメーターを指定せずに `rsh` コマンドを使用してください。*Command* パラメーターを指定しないと、`rsh` コマンドは、代わりに `rlogin` コマンドを実行します。

### アクセス・ファイル

`-l` フラグを指定しないと、ローカル・ユーザー名がリモート・ホスト側で使用されます。`-l User` を入力すると、指定したユーザー名がリモート・ホスト側で使用されます。

### 標準の認証の使用

次の条件のうち、少なくとも 1 つが満たされた場合にのみ、リモート・ホストへのアクセスが可能になります。

- ローカル・ユーザー ID が root ユーザーでなく、またローカル・ホストの名前が、リモートの `/etc/hosts.equiv` ファイルに同等のホストとしてリストされています。
- ローカル・ユーザー ID が root ユーザーの場合、または `/etc/hosts.equiv` の検査が正常終了でない場合には、リモート・ユーザーのホーム・ディレクトリーには、ローカル・ホストとユーザー名のリストを含む `$HOME/.rhosts` ファイルが必要です。

ユーザーは `$HOME/.rhosts` ファイルに対して、許可を任意に設定できますが、`.rhosts` ファイルの許可は 600 (オーナーだけが読み取りおよび書き込み可能) に設定することをお勧めします。

上記の条件の他に、リモート・ユーザーのアカウントにパスワードが定義されていなければ、`rsh` コマンドを使用してリモート・ホストにアクセスできます。ただし、セキュリティー上の理由で、すべてのユーザー・アカウント上でパスワードを使用することをお勧めします。

## Kerberos 5 認証の場合

次の条件のすべてが満たされた場合にのみ、リモート・ホストへのアクセスが可能になります。

- ローカル・ユーザーに現在の DCE 証明書が与えられている。
- ローカル・システムとリモート・システムが、Kerberos 5 認証に対して構成されている (一部のリモート・システムの場合、これは不要のことがある。デーモンが `klogin` ポートを `listen` していることが必要)。
- リモート・システムが DCE 証明書をリモート・アカウントへのアクセスに充分であるとして受け入れる。追加情報については、`kvalid_user` 関数を参照のこと。

## リモート・コマンドの実行

リモート・コマンドの実行中に、割り込み (Interrupt)、完了 (Terminate)、あるいは終了 (Quit) のキー・シーケンスを押すと、対応するシグナルがリモート・プロセスに送信されます。ただし、停止 (Stop) のキー・シーケンスを押すと、ローカル・プロセスだけが停止します。通常、リモート・コマンドが終了すると、ローカルの `rsh` プロセスは終了します。

シェル・メタキャラクターがリモート・ホスト上で解釈されるようにするには、メタキャラクターを " " (二重引用符) で囲んでください。これ以外の場合には、メタキャラクターがローカル・シェルにより解釈されます。

`rsh` コマンドを使用すると、(ユーザーが書き込み許可を持っている) パスへのリンクを作成することができます。このとき、`HostName` パラメーターで指定したホスト名がリンク名として使用されます。以下に例を示します。

```
ln -s /usr/bin/rsh HostName
```

このリンクの設定が完了すれば、`HostName` パラメーター、およびコマンド行から `Command` パラメーターで指定したコマンドを指定できます。また、このコマンドは `rsh` コマンドによってリモート・ホスト上でリモートで実行されます。この構文は次のとおりです。

```
HostName Command
```

例えば、リモート・ホスト `opus` にリンクしているときに、`date` コマンドを実行したい場合は、以下のように入力します。

```
opus date
```

**-l** `User` フラグが指定できないため、ローカル・ユーザーがリモート・ホスト上にユーザー・アカウントを所有している場合にしか、リモート・コマンドは正しく実行されません。これ以外の場合には、`rsh` コマンドにより、`Login incorrect` というエラー・メッセージが戻されます。コマンドを指定せずに `HostName`

パラメーターを指定すると、**rsh** コマンドは **rlogin** コマンドを呼び出し、このコマンドにより、リモート・ホストにログインすることができます。この場合にも、ログインを正常に行うには、ローカル・ユーザーがリモート・ホスト上にユーザー・アカウントを所有していなければなりません。

## フラグ

- a** リモート・コマンドの標準エラーを指示は、標準出力と同様です。任意のシグナルをリモート・プロセスに送信するサービスは提供されません。
- f** これを指定すると、証明書が転送されます。このフラグは、Kerberos 5 が現在の認証メソッドでない場合は無視されます。認証は、現在の DCE 証明書に転送可能のマークが付けられていない場合には無視されます。
- F** これを指定すると、証明書が転送されます。さらに、リモート・システムの証明書に転送可能のマークが付けられます (これによって、別のリモート・システムにそれらの証明書を渡すことができるようになります)。このフラグは、Kerberos 5 が現在の認証メソッドでない場合は無視されます。認証は、現在の DCE 証明書に転送可能のマークが付けられていない場合には無視されます。
- k realm** これを使用することで、ユーザーは、リモート端末のレルムを指定できます。ただし、そのレルムが DCE セルと同意である場合に限りです。このフラグは、Kerberos 5 が現在の認証メソッドでない場合は無視されます。
- l User** **rsh** コマンドが、ローカル・ユーザー名ではなく、*User* 変数によって指定されたユーザーとしてリモート・ホストにログインすることを指定します。このフラグを指定しないと、ローカルのユーザー名とリモートのユーザー名が同一になります。
- n** **rsh** コマンドが標準入力からの読み取りを実行しないように指定します。
- S** セキュア・オプション。標準エラー接続のリモート IP アドレスが、標準出力接続と同じになるように強制します。
- u** 標準の AIX 認証のみを使用します。

## セキュリティ

次の条件のうち、少なくとも 1 つが満たされた場合にのみ、リモート・ホストへのアクセスが可能になります。

- ローカル・ユーザー ID が認証データベース内でプリンシパルとしてリストされ、**kinit** を実行してチケットを獲得していた場合。
- **\$HOME/.klogin** ファイルがある場合は、そのファイルがターゲット・システム上のローカル・ユーザーの **\$HOME** ディレクトリーにある場合。ローカル・ユーザーは、さらにこのアカウントに **rsh** を許可されたユーザーまたはサービスとしてリストされなければなりません。このファイルは、ローカル **.rhosts** ファイルと同等の機能を実行します。このファイル内の各行には、*principal.instance@realm* 形式のプリンシパルがなければなりません。親ユーザーが **.klogin** 内で命名されたプリンシパルの 1 つとして認証されている場合は、アカウントへのアクセスが認可されます。アカウントのオーナーは **.klogin** ファイルがない場合にアクセスが認可されます。

セキュリティの理由から、**\$HOME/.klogin** ファイルはリモート・ユーザーが所有し、**.klogin** への読み取り書き込みアクセス (許可 = 600) を持つのは、AIX オーナー ID だけにする必要があります。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0** 正常終了。
- >0** エラーが発生しました。

## 例

以下の例にあるローカル・ホスト、host1 は、リモート・ホスト、host2 側の `/etc/hosts.equiv` ファイルにリストされています。

1. リモート・ホスト上の空きディスク・スペース容量を検査するには、次のように入力します。

```
rsh host2 df
```

host2 上の空きディスク・スペース容量はローカル・システムに表示されます。

2. リモート・ファイルのリモート・ホスト上の別のファイルに追加するには、`>>` メタキャラクターを引用符で囲み次のように入力します。

```
rsh host2 cat test1 ">>" test2
```

ファイル test1 が、リモート・ホスト host2 上の test2 に追加されます。

3. リモート・ホスト上のリモート・ファイルを、ローカル・ファイルに追加するには、引用符を省略し、次のように入力します。

```
rsh host2 cat test2 >> test3
```

host2 上のリモート・ファイル test2 が、ローカル・ファイル test3 に追加されます。

4. リモート・ファイルをローカル・ファイルに追加し、リモート・ホストでリモート・ユーザーの許可を使用するには、次のように入力します。

```
rsh host2 -l jane cat test4 >> test5
```

リモート・ファイル test4 が、リモート・ホストにおいてユーザー jane の許可で、ローカル・ファイル test5 に追加されます。

5. この例は、ターゲットとサーバーの両方で認証が Kerberos 4 の場合に、root ユーザーがリモート・ホスト上で **rcp** を出す方法を示しています。root ユーザーは認証データベースに登録され、ローカル・ホスト上に **kinit** を出しておく必要があります。コマンドはローカル・ホストで出され、ファイル stuff をノード r05n07 から SP 上の r05n05 へコピーします。

```
/usr/lpp/ssp/rcmd/bin/rsh r05n07 'export KRBTKTFILE=/tmp/rcmdtkkt$$; \  
/usr/lpp/ssp/rcmd/bin/rcmdtgt; \  
/usr/lpp/ssp/rcmd/bin/rcp /tmp/stuff r05n05:/tmp/stuff;'
```

root ユーザーは、KRBTKTFILE 環境変数を一時チケット・キャッシュ・ファイル名に設定してから、**rcmdtgt** コマンドを出してサービス・チケットを獲得します。**rcp** はサービス・チケットを使用して、ホスト r05n07 からホスト r05n05 へ認証します。

## ファイル

**\$HOME/.klogin**

ローカル・ユーザー・アカウントを使用できるリモート・ユーザーを指定します。

**/usr/lpp/ssp/rcmd/bin/rsh**

SP Kerberos 4 **rsh** ルーチンを (利用可能な場合に) 呼び出す AIX セキュア **/usr/bin/rsh** へのリンク。

**/usr/lpp/ssp/rcmd/bin/remsh**

SP Kerberos 4 **rsh** ルーチンを (利用可能な場合に) 呼び出す AIX セキュア **/usr/bin/rsh** へのリンク。



## 前提条件情報

概要については、「IBM Parallel System Support Programs for AIX: Administration Guide」のセキュリティに関する章を参照してください。この資料には、Web サイト [http://www.rs6000.ibm.com/resource/aix\\_resource](http://www.rs6000.ibm.com/resource/aix_resource) でアクセスできます。

Kerberos の追加情報については、「IBM Parallel System Support Programs for AIX: Command and Technical Reference」の RS/6000 SP Files and Other Technical Information のセクションを参照してください。この資料には、Web サイト [http://www.rs6000.ibm.com/resource/aix\\_resource](http://www.rs6000.ibm.com/resource/aix_resource) でアクセスできます。

## 関連情報

**ftp** コマンド、**rcp** コマンド、**rexec** コマンド、**rlogin** コマンド、**telnet**、**tn**、または **tn3270** コマンド。

SP コマンド: **k4init**、**k4list**、**k4destroy**、**lsauthpar**、**chauthpar**、**Kerberos**。

環境変数 **KRBTKFILE**。

**rshd** および **krshd** デーモン。

**kvalid\_user** 関数。

**hosts.equiv** ファイル・フォーマット、**.rhosts** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ の通信およびネットワークのセクション。

ネットワークおよびコミュニケーションのマネージ の認証およびセキュア **rcmd** のセクション。

---

## rshd デーモン

### 目的

リモート・コマンドを実行するためにサーバー機能を提供します。

### 構文

注: **rshd** デーモンは通常、**inetd** デーモンによって開始されます。また、**SRC** コマンドを使用してコマンド行から制御することもできます。

```
/usr/sbin/rshd [ -c ] [ -s ] [ -p ]
```

### 説明

**/usr/sbin/rshd** デーモンは、**rcp** コマンドおよび **rsh** コマンドのサーバーです。**rshd** デーモンを使用すると、シェル・コマンドをリモートで実行できます。これらのコマンドは、トラステッド・ホスト上の優先ソケットからの要求に基づいています。このようなシェル・コマンドには、ユーザーの確認が必要です。**rshd** デーモンは、**/etc/services** ファイルに定義されているソケットでこの確認を待ちます。

**rshd** デーモンの変更は、System Management Interface Tool (SMIT) またはシステム・リソース・コントローラー (SRC) を使用して、**/etc/inetd.conf** ファイルまたは **/etc/services** ファイルを編集することによって行うことができます。コマンド行で **rshd** と入力することはお勧めできません。**rshd** デーモンは、**/etc/inetd.conf** ファイル内にコメントされていないときに、デフォルトで始動されます。



**inetd** デーモンは、 **/etc/inetd.conf** ファイルと **/etc/services** ファイルの情報を取得します。

**/etc/inetd.conf** ファイルまたは **/etc/services** ファイルを変更したら、 **refresh -s inetd** または **kill -1 InetdPID** コマンドを実行して、構成ファイルを変更したことを **inetd** デーモンに通知します。

## サービス要求プロトコル

**rshd** デーモンは、サービス要求を受け取るときに、次のプロトコルを開始します。

1. **rshd** デーモンは、要求の送信元ポート番号を検査します。ポート番号が 512 から 1023 の範囲内にならない場合、**rshd** デーモンは接続を終了します。
2. **rshd** デーモンが、ソケットから、 **null** バイトまでの文字を読み取ります。読み取られた文字列は、ASCII 番号 (基数 10) として解釈されます。この数値がゼロ以外の場合、**rshd** デーモンは、標準エラーとして使用される 2 次局ストリームのポート番号としてこれを解釈します。クライアント・ホスト上の指定のポートに、2 番目の接続が作成されます。ローカル・ホスト上のソース・ポートも、512 から 1023 の範囲内です。
3. **rshd** デーモンは、最初の接続要求の送信元アドレスを使用して、クライアント・ホストの名前を判別します。名前を判別できない場合、**rshd** デーモンは、クライアント・ホストのアドレスのドット 10 進数を使用します。
4. **rshd** デーモンは、最初のソケットから、次の情報を検索します。
  - クライアント・ホスト上のユーザーのユーザー名として解釈される最大 16 バイトの **null** 終止文字列。
  - ローカル・サーバー・ホスト上で使用されるユーザー名として解釈される最大 16 バイトの **null** 終止文字列。
  - ローカル・サーバー・ホスト上のシェルに渡されるコマンド行として解釈されるその他の **null** 終止文字列。
5. **rshd** デーモンは、次のステップを使用して、ユーザーの妥当性検査を行おうとします。
  - a. **rshd** デーモンは、 **/etc/passwd** および **/etc/security/passwd** ファイル内のローカル・ユーザー名を検索し、パスワード (したがってユーザー) が期限切れかどうかを検査します。また、暗号化パスワード文字列が有効かどうかを検査し、 (**chdir** サブルーチンを使用して) ホーム・ディレクトリーへの切り替えを試みます。ファイル内の調査またはディレクトリーの変更が正しく行われなかった場合、**rshd** デーモンは接続を終了します。
  - b. ローカル・ユーザー ID がゼロ以外の値であれば、**rshd** デーモンは、**/etc/hosts.equiv** ファイルを検索して、クライアント・ワークステーションの名前がリストされているかどうかを確認します。クライアント・ワークステーションが同等のホストとしてリストされている場合、**rshd** デーモンは、ユーザーの妥当性検査を行います。
  - c. **\$HOME/.rhosts** ファイルが存在する場合、**rshd** デーモンは、**.rhosts** ファイルを検査して、ユーザーを認証しようとしています。
  - d. **\$HOME/.rhosts** の認証が正しく行われなかった場合、またはクライアント・ホストが同等のホストでない場合、**rshd** デーモンは接続を終了します。
6. **rshd** デーモンはユーザーの妥当性を確認した後、最初の接続に **null** バイトを戻し、ユーザーのローカル・ログイン・シェルにコマンド行を渡します。次に、シェルでは、**rshd** デーモンにより確立されたネットワーク接続を継承します。

**rshd** デーモンは、System Management Interface Tool (SMIT) を使用するか、あるいは **/etc/inetd.conf** ファイルを変更することにより制御する必要があります。コマンド行で **rshd** と入力することはお勧めできません。

## システム・リソース・コントローラーを使用した rshd デーモンの操作

**rshd** デーモンは、システム・リソース・コントローラー (SRC) のサブシステムである **inetd** デーモンのサブサーバーです。 **rshd** デーモンは、**tcpip** SRC サブシステム・グループのメンバーです。このデーモンは、デフォルトでは **/etc/inetd.conf** ファイルの定義により使用可能となり、また、以下の SRC コマンドを使用して操作することができます。

|                 |                                        |
|-----------------|----------------------------------------|
| <b>startsrc</b> | サブシステム、サブシステムのグループ、あるいはサブサーバーを開始します。   |
| <b>stopsrc</b>  | サブシステム、サブシステムのグループ、あるいはサブサーバーを停止します。   |
| <b>lssrc</b>    | サブシステム、サブシステム・グループ、またはサブサーバーの状況を取得します。 |

## フラグ

- c ホスト名ルックアップの健全性チェックを抑制します。
- p 非対話式モードで **rsh** コマンドを発行する場合はいつも、**.profile** ファイルを実行してください。このフラグを指定しないと、非対話式モードでの **rsh** コマンドのケースでは、**.profile** ファイルは稼働しません。
- s ソケット・レベルのデバッグをオンにします。

## セキュリティ

**rshd** デーモンは、**rsh** というサービス名を持つ、PAM 使用可能アプリケーションです。認証に PAM を使用するシステム規模の構成は、**/etc/security/login.cfg** の **usw** スタンザにある **auth\_type** 属性の値を、root ユーザーと同じ **PAM\_AUTH** に変更することにより、設定されます。

PAM が使用可能であるときに使用される認証メカニズムは、**/etc/pam.conf** 内の **rsh** サービスの構成によって決まります。 **rshd** デーモンは、**auth** モジュール・タイプ、**account** モジュール・タイプ、および **session** モジュール・タイプの **/etc/pam.conf** エントリーを必要とします。以下にリストされるのは、**rsh** サービスにおける **/etc/pam.conf** 内の推奨される構成です。

```
#
# AIX rsh configuration
#
rsh auth      sufficient  /usr/lib/security/pam_rhosts_auth
rsh account   required    /usr/lib/security/pam_aix
rsh session   required    /usr/lib/security/pam_aix
```

## 例

注: **rshd** デーモンの引数は、SMIT を使用するか、または **/etc/inetd.conf** ファイルを編集することによって指定できます。

1. **rshd** デーモンを始動するには、次のように入力します。

```
startsrc -t shell
```

このコマンドは **rshd** サブサーバーを始動します。

2. **rshd** デーモンを停止するには、次のように入力します。

```
stopsrc -t shell
```

このコマンドにより、保留状態の接続がすべて開始され、既存の接続が完了します。ただし、新しい接続の開始は行われません。

3. **rshd** デーモンおよびすべての **rshd** 接続を強制的に停止するには、以下のように入力します。

```
stopsrc -t -f shell
```

このコマンドにより、保留状態の接続と既存の接続がすべて即座に終了します。

4. **rshd** デーモンに関する簡潔なフォーマットの状況報告を表示するには、以下のように入力します。

```
lssrc -t shell
```

このコマンドは、デーモンの名前、プロセス ID、および状態 (アクティブか非アクティブか) を戻します。

## 関連情報

**kill** コマンド、**lssrc** コマンド、**refresh** コマンド、**rsh** コマンド、**startsrc** コマンド、**stopsrc** コマンド。

**inetd** デーモン。

**\$HOME/.rhosts** ファイル・フォーマット、**/etc/hosts.equiv** ファイル・フォーマット、**/etc/inetd.conf** ファイル・フォーマット、**/etc/services** ファイル・フォーマット。

**rcmd** サブルーチン。

ネットワークおよびコミュニケーションのマネージ の TCP/IP デーモンのセクション。

---

## rstatd デーモン

### 目的

カーネルから入手したパフォーマンス統計情報を戻します。

### 構文

```
/usr/sbin/rpc.rstatd
```

### 説明

**rstatd** デーモンは、カーネルから得たパフォーマンス統計情報を戻すサーバーです。**rstatd** デーモンは通常、**inetd** デーモンによって始動されます。

### ファイル

**/etc/inetd.conf**  
**/etc/services**

RPC デーモンおよび他の TCP/IP デーモンを開始する TCP/IP 構成ファイル。  
インターネットで利用できる各サーバーのエントリが入っています。

## 関連情報

**nfsstat** コマンド。

**inetd** デーモン。

ネットワークおよびコミュニケーションのマネージ のネットワーク・ファイルシステム (NFS) の概要のセクション。

ネットワークおよびコミュニケーションのマネージ の NFS コマンドのリストのセクション。

---

## rtl\_enable コマンド

### 目的

実行時リンカーで使用できるように共有オブジェクトを再リンクします。このコマンドは、AIX 4.2 以降にだけ適用されます。

### 構文

```
rtl_enable [ -R | -o Name ] [ -l ] [ -s ] File [ ldFlag ... ] [ -F ObjsLibs ... ]
```

### 説明

**rtl\_enable** コマンドは、**-G** フラグを持つ、モジュールまたはモジュールを含むアーカイブに再リンクして、実行時リンクを可能にします。モジュールは、ローダー・セクションを含む XCOFF ファイルです。共有オブジェクトは、XCOFF ヘッダーに **F\_SHROBJ** フラグを設定したモジュールです。

最も簡単なフォーマットでは、**rtl\_enable** コマンドは **File.new** という名前の新規ファイルを作成します。**File** がモジュールであれば、**File.new** は同じ種類のモジュールになります。**File** がアーカイブであれば、**File.new** メンバーが **File** のメンバーと同じ名前を持つアーカイブになります。**rtl\_enable** コマンドは新規アーカイブのモジュールに再リンクして、実行時リンクを可能にします。**rtl\_enable** コマンドは、他のメンバーを変更しないで出力ファイルに保管します。

**rtl\_enable** コマンドは、**File** (またはそのメンバー) のローダー・セクションを使用して、インポート・ファイルおよびエクスポート・ファイルを作成し、**libpath** 情報を判別し、エントリー・ポイントを決定します。

### フラグ

- |                              |                                                                                                                                                                                                                                                                      |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-F</b> <i>ObjsLibs...</i> | <i>ObjsLibs</i> を、生成された <b>ld</b> コマンドの先頭に追加します。 <i>ObjsLibs</i> パラメーターは、オブジェクト・ファイルかまたはライブラリー ( <b>ld</b> command's <b>-l</b> (L の小文字) フラグで指定) です。アーカイブを使用可能にする場合は、そのアーカイブのすべての共有オブジェクトについて、 <i>ObjsLibs</i> を <b>ld</b> コマンドに追加します。                                |
| <b>-l</b>                    | (L の小文字) インポート・ファイルおよびエクスポート・ファイルは削除せずに現行ディレクトリーに入れたままにします。インポート・ファイルは接尾部 <b>.imp</b> を持ち、エクスポート・ファイルは接尾部 <b>.exp</b> を持ちます。 <b>File</b> がモジュールである場合は、 <b>rtl_enable</b> コマンドによって入力ファイル名に接尾部が追加されます。 <b>File</b> がアーカイブである場合は、このコマンドによって、モジュールであるメンバーの名前に接尾部が追加されます。 |
| <b>-o</b> <i>Name</i>        | <b>File.new</b> の代わりに代替出力ファイル名を指定します。このフラグを <b>-R</b> フラグと一緒に使用しないでください。                                                                                                                                                                                             |
| <b>-R</b>                    | 新規ファイルを作成する代わりに入力ファイルを置換します。エラーが発生すると、入力ファイルは上書きされません。このフラグを <b>-o</b> フラグと一緒に使用しないでください。                                                                                                                                                                            |

- s** 新規出力ファイルまたは新規アーカイブを作成するときを使用することのできる現行ディレクトリーに、コマンドのスクリプトを生成しますが、再リンクはしません。これによってスクリプトの名前は **Base.sh** となります。ここで **Base** は、入力ファイルから接尾部を取り除いたベース名です。生成されたインポート・ファイルおよびエクスポート・ファイルも同様に現行ディレクトリーに書き込まれます。スクリプト、インポート・ファイル、およびエクスポート・ファイルを変更して、出力オブジェクトをカスタマイズすることができます。

## パラメーター

- File** 入力ファイルを指定します。  
**ldFlag...** 指定した **ld** コマンド・フラグを、生成した **ld** コマンドの最後にコピーし、デフォルト・オプションを上書きします。  
**注:** 出力ファイルに名前を付けるときに、**ldFlag** パラメーターの **-o** フラグを使用しないでください。代替出力ファイル名を指定するには、**rtl\_enable** コマンドの **-o Name** フラグを使用してください。

## 終了状況

このコマンドは、以下の終了値を戻します。

- 0** 正常終了。  
**>0** エラーが発生しました。

**注:** エラーによっては、出力ファイルが作成されているものもあります。

## セキュリティ

アクセス制御: 任意のユーザー

イベントの監査: N/A

## 例

実行時にリンク可能な **libc.a** の新規バージョンを作成するには、以下のように入力します。

1. 実行時バージョンのディレクトリーを作成するために、次のように入力します。

```
mkdir /tmp/rtllibs
```

2. **/tmp/rtllibs** を、ユーザーの現行ディレクトリーにするために次のように入力します。

```
cd /tmp/rtllibs
```

3. 同じ名前でも **libc.a** の実行時バージョンを作成するために、次のように入力します。

```
rtl_enable -o libc.a /lib/libc.a
```

プログラムにリンクしたときに **libc.a** のこのバージョンを使用するには、**ld** コマンドを持つ **-L /tmp/rtllibs** を使用します。

## ファイル

- /usr/bin/rtl\_enable** **rtl\_enable** コマンドが入っています。これは、**/usr/ccs/bin/rtl\_enable** へのシンボリック・リンクです。

## 関連情報

コマンド: `ld`。

AIX 5L バージョン 5.3 プログラミングの一般概念: プログラムの作成およびデバッグの共用オブジェクトと実行時リンクのセクション。

---

## runacct コマンド

### 目的

日次アカウントिंगを実行します。

### 構文

```
/usr/sbin/acct/runacct [ mmdd [ State ] ]
```

### 説明

**runacct** コマンドは、主な日次アカウントिंगのシェル・プロシージャです。通常 **runacct** コマンドは、**cron** デーモンによって開始されると、現行日の接続、料金、ディスク、キューイング・システム (プリンター)、およびプロセスのアカウントिंग・データ・ファイル进行处理して、バイナリー日次レポート **/var/adm/acct/nite(x)/dayacct** を作成します。また、**runacct** コマンドは、ASCII 日次レポート **/var/adm/acct/sum(x)/rprtmdd** を作成する **prdaily** プロシージャまたは請求のために、要約ファイルを準備します。

**acctmerg** コマンドは、アカウントング期間の累積要約レポート **/var/adm/acct/sum(x)/tacct** に **dayacct** レポートを追加します。**tacct** レポートは、月次レポート **/var/adm/acct/fiscal(x)** を作成するために、**monacct** コマンドが使用します。

このコマンドには、**runacct** プロシージャを再始動させる場合にキーボードから入力する必要がある 2 つのパラメーターがあります。日付パラメーター、*mmdd* を使用して、アカウントिंगをどの日および月に対して再実行したいかを指定できます。また、管理権限を持っているユーザーは、*State* パラメーターを使用して、どの状態からでも、**runacct** プロシージャを再始動できます。**runacct** プロシージャの再始動、および障害からのリカバリーの詳細を知りたい場合は、**runacct** プロシージャの再始動を参照してください。

**runacct** コマンドは、実行時エラーが発生した場合に、アクティブ・アカウント・ファイルと要約ファイルを保護し、説明メッセージを **/var/adm/acct/nite(x)/active** ファイルに書き込むことによって、その進行状況を記録します。**runacct** プロシージャにエラーが発生すると、メールが root ユーザーと adm ユーザーに送信され、プロシージャが終了します。

**runacct** プロシージャでは、ディレクトリー **/var/adm/acct/nite(x)** 内に 2 つの一時ファイル **lock** と **lock1** が作成されます。**runacct** プロシージャは、これらの一時ファイルを使用して、このプロシージャに対して同時に 2 つのコールが行われないようにします。また、このプロシージャでは、1 日当たり 2 回以上起動されることがないように、(同じディレクトリー内にある) **lastdate** ファイルが使用されません。

**runacct** コマンドは、処理を個別の再始動可能な状態に分割します。このコマンドは、1 つの状態を完了させると、次の状態の名前を **/var/adm/acct/nite/state** ファイルに書き込みます。**runacct** プロシージャ



ーでは、下記の順序で、さまざまな状態が処理されます。

| 状態                | アクション                                                                                                                 |
|-------------------|-----------------------------------------------------------------------------------------------------------------------|
| <b>SETUP</b>      | アクティブ・アカウントिंग・ファイルを作業ファイルへ移動させて、アクティブ・ファイルを再始動します。                                                                   |
| <b>WTMPFIX</b>    | <b>wtmp</b> ファイルの整合性を検査し、必要な場合は、日付変更を修正します。                                                                           |
| <b>CONNECT1</b>   | <b>acctcon1</b> コマンドを呼び出して、接続セッション・レコードを作成します。                                                                        |
| <b>CONNECT2</b>   | 接続セッション・レコードを、合計アカウントING・レコードに変換します ( <b>tacct.h</b> フォーマット)。                                                         |
| <b>PROCESS</b>    | プロセス・アカウントING・レコードを、合計アカウントING・レコードに変換します ( <b>tacct.h</b> フォーマット)。                                                   |
| <b>MERGE</b>      | 接続アカウントING・レコードとプロセス合計アカウントING・レコードを組み合わせます。                                                                          |
| <b>FEES</b>       | <b>chargefee</b> コマンドの出力を合計アカウントING・レコードに変換し ( <b>tacct.h</b> フォーマットで)、それらを、接続およびプロセス合計アカウントING・レコードと組み合わせます。         |
| <b>DISK</b>       | ディスク・アカウントING・レコードを、接続、プロセス、および料金合計アカウントING・レコードと組み合わせます。                                                             |
| <b>QUEUEACCT</b>  | キュー (プリンター) アカウントING・レコードをソートして、それらを合計アカウントING・レコードに変換し ( <b>tacct.h</b> フォーマット)、さらに他の合計アカウントING・レコードと組み合わせます。        |
| <b>MERGETACCT</b> | <b>daytacct</b> レポート・ファイル内の日次合計アカウントING・レコードを、<br><b>/var/adm/acct/sum/tacct</b> レポート・ファイル内の要約合計アカウントING・レコードとマージします。 |
| <b>CMS</b>        | <b>/var/adm/acct/sum(x)/cms</b> ファイル内でコマンド要約を作成します。                                                                   |
| <b>USEREXIT</b>   | <b>/var/adm/siteacct</b> シェル・ファイルがある場合は、それをこの時点でコールして、サイト依存処理を実行します。                                                  |
| <b>CLEANUP</b>    | 一時ファイルを削除して終了します。                                                                                                     |

## runacct プロシーチャーの再始動

障害が発生した後に、**runacct** コマンドを再始動するには、まず **/var/adm/acct/nite(x)/active** ファイルで診断メッセージを調べ、**pacct** または **wtmp** などの損傷データ・ファイルを修正します。そして、**runacct** コマンドを再始動する前に、**lock** ファイルと **lastdate** ファイル (いずれも **/var/adm/acct/nite(x)** ディレクトリーに入っている) を除去します。**runacct** コマンドを再始動させるときは、**mmdd** パラメーターを指定しなければなりません。このパラメーターでは、**runacct** コマンドがアカウントINGをどの月日に関して再実行するのかを指定します。**runacct** プロシーチャーでは、**/var/adm/acct/nite(x)/statefile** ファイルを読み取ることによって、処理用のエントリー・ポイントが決定されます。このデフォルト・アクションを無効にするには、**runacct** コマンド行で希望する **state** を指定します。

通常、**SETUP state** で **runacct** コマンドを再始動するのは、有効な方法とはいえません。それよりも、セットアップ・アクションを手動で実行し、下記のように、**WTMPFIX** 状態を指定してアカウントINGを再始動してください。

```
/usr/lib/acct/runacct mmdd WTMPFIX
```

**runacct** コマンドが **PROCESS** 状態で正しく動作しない場合は、最後の **ptacct** ファイルを除去します。不完全なものになるからです。



## フラグ

- X 最初の 8 文字に切り捨てることをせずに、各ユーザー名の使用可能なすべての文字を処理します。また、-X フラグを使用すると、**runacct** コマンドとそれが呼び出すすべてのコマンドが、**/var/adm/acct/sum** および **/var/adm/acct/nite** ディレクトリーではなく、**/var/adm/acct/sumx** および **/var/adm/acct/nitex** ディレクトリーを使用します。

## セキュリティ

アクセス制御: このコマンドは、**adm** グループのメンバーのみに実行 (x) アクセス権限を与えます。

## 例

1. システム・リソースに対して日次アカウントिंग手順を始動するには、下記のコマンド行を **crontab** ファイルに追加し、**runacct** コマンドが、**cron** デーモンによって自動的に実行されるようにします。

```
0 4 * * 1-6 /usr/sbin/acct/runacct 2> \  
/var/adm/acct/nite/accterr
```

長いユーザー名のサポートを指定して、日次アカウントING手順を開始するには、**crontab** ファイルに次の行を追加します。

```
0 4 * * 1-6 /usr/sbin/acct/runacct -X 2> \  
/var/adm/acct/nitex/accterr
```

この例は、**cron** デーモンが読み取って実行する命令を示しています。**runacct** コマンドは、毎月曜日から土曜日まで (1-6) の間の午前 4 時 (04) に実行され、すべての標準エラー出力 (2>) を、**/var/adm/acct/nite(x)/accterr** ファイルに書き込みます。このコマンドは、**cron** デーモンに通常与えられるアカウントING命令の 1 つにすぎません。典型的な **cron** アカウント・エントリーの詳細は、オペレーティング・システムおよびデバイスのマネージのアカウントING・サブシステムの設定のセクションを参照してください。

2. システム・リソースに対して、コマンド行 (**runacct** コマンドを始動させる) から日アカウントING手順を始動させるには、下記のように入力します。

```
nohup /usr/sbin/acct/runacct 2> \  
/var/adm/acct/nite/accterr &
```

**cron** デーモンによって **runacct** プロシーチャーを自動的に始動させることをお勧めしますが (例 1 を参照)、キーボードからコマンドを入力してもかまいません。**runacct** コマンドはバックグラウンド (&) で実行され、すべての INTERRUPT シグナルおよび QUIT シグナルは無視されます (**nohup** コマンド)。また、すべての標準エラー出力 (2>) は、**/var/adm/acct/nite/accterr** ファイルに書き込まれます。

3. 特定の日付に対してシステム・アカウントING手順を再始動させるには、下記のようなコマンドを入力します。

```
nohup /usr/sbin/acct/runacct 0601 2>> \  
/var/adm/acct/nite/accterr &
```

この例では、6 月 1 日 (0601) という日に関して、**runacct** コマンドが再始動されます。**runacct** コマンドは、ファイル **/var/adm/acct/nite(x)/statefile** を読み取って、開始するときの状態を探します。**runacct** コマンドはバックグラウンド (&) で実行されて、すべての INTERRUPT シグナルおよび QUIT シグナルが無視されます (**nohup**)。また、標準エラー出力 (2) が **/var/adm/acct/nite(x)/accterr** ファイルの最後 (>>) に追加されます。

4. 特定の状態で、特定の日付に対するシステム・アカウントिंग手順を再始動させるには、下記のようなコマンドを入力します。

```
nohup /usr/sbin/acct/runacct 0601 MERGE 2>> \  
/var/adm/acct/nite(x)/accterr &
```

この例では、6月1日(0601)という日に関して、**runacct** コマンドが始動され、MERGE 状態で開始します。**runacct** コマンドはバックグラウンド (&) で実行されて、すべての INTERRUPT シグナルおよび QUIT シグナルが無視されます (**nohup** コマンド)。また、標準エラー出力 (2) が **/var/adm/acct/nite(x)/accterr** ファイルの最後 (>>) に追加されます。

## ファイル

|                                          |                                    |
|------------------------------------------|------------------------------------|
| <b>/var/adm/wtmp</b>                     | ログイン/ログオフ・履歴・ファイル。                 |
| <b>/var/adm/pacct*</b>                   | プロセス・アカウントング・ファイル。                 |
| <b>/var/adm/acct/nite(x)/dayacct</b>     | ディスク使用状況アカウントング・ファイル。              |
| <b>/var/adm/qacct</b>                    | アクティブ・キュー・アカウントング・ファイル。            |
| <b>/var/adm/fee</b>                      | ユーザーに課される料金のレコード。                  |
| <b>/var/adm/acct/sum(x)*</b>             | コマンド・ファイルと合計アカウントング要約ファイル。         |
| <b>/var/adm/acct/nite(x)/ptacct*.mmd</b> | <b>pacct</b> ファイルの連結バージョン。         |
| <b>/var/adm/acct/nite(x)/active</b>      | <b>runacct</b> メッセージ・ファイル。         |
| <b>/var/adm/acct/nite(x)/lock*</b>       | <b>runacct</b> コマンドの同時呼び出しを防止します。  |
| <b>/var/adm/acct/nite(x)/lastdate</b>    | <b>runacct</b> が実行された最後の日付が入っています。 |
| <b>/var/adm/acct/nite(x)/statefile</b>   | 処理すべき現行状態が入っています。                  |

## 関連情報

**acctcms** コマンド、**acctcom** コマンド、**acctcon1** または **acctcon2** コマンド、**acctmerge** コマンド、**acctprc1**、**acctprc2**、または **accton** コマンド、**crontab** コマンド、**fwtmp** コマンド。

**cron** デーモン。

**acct** サブルーチン。

**acct** ファイル・フォーマット、**utmp**、**wtmp**、**failedlogin** ファイル・フォーマット。

アカウントング・システム、日次および月次のレポートの準備、およびアカウントング・ファイルについての詳細は、オペレーティング・システムおよびデバイスのマネージのシステム・アカウントングのセクションを参照してください。

オペレーティング・システムおよびデバイスのマネージのアカウントング・サブシステムの設定のセクションで、アカウントング・システムの確立のための必要なステップが説明されています。

自動的に実行するコマンドまたはキーボードから入力できるコマンドのリストについては、オペレーティング・システムおよびデバイスのマネージのアカウントング・コマンドを参照してください。

---

## runact コマンド

### 目的

リソース・クラスに対するアクションを実行します。

## 構文

```
runact -s "selection_string" [ -N { node_file | "-" } ] [-f resource_data_input_file] [-l | -t | -d | -D delimiter] [-x] [-h] [-TV] resource_class action [in_element=value...] [rsp_element...]
```

```
runact -r [-f resource_data_input_file] [-l | -t | -d | -D delimiter] [-x] [-h] [-TV] resource_handle action [in_element=value...] [rsp_element...]
```

```
runact -c [-f resource_data_input_file] [-n node_name] [-l | -t | -d | -D delimiter] [-x] [-h] [-TV] resource_class action [in_element=value...] [rsp_element...]
```

```
runact -C domain_name... [-f resource_data_input_file] [-l | -t | -d | -D delimiter] [-x] [-h] [-TV] resource_class action [in_element=value...] [rsp_element...]
```

## 説明

**runact** コマンドは、指定のリソース・クラス上で指定のアクションを RMC サブシステムが実行するように要求します。

*selection\_string* に複数のノード名を指定する代わりに、**-N** *node\_file* フラグを使用して、ノード名がファイルにあることを指示できます。標準入力からノード名を読み取るには、**-N** "-" を使用します。

このコマンド実行前に **lsactdef** コマンドを使用して、このリソース・クラスによってサポートされるリソース・クラス・アクションのリストを入手してください。また、**lsactdef** コマンドを使用することによって、アクション起動時に指定する必要がある必須入力アクション要素のリストも入手してください。

**lsactdef** コマンドを使用すると、各入力要素のデータ・タイプを調べることもできます。各入力に対して指定する値は、そのデータ・タイプに一致したものでなければなりません。

## フラグ

**-c** リソース・クラスに対してアクションを起動します。

**-C** *domain\_name...*

管理サーバーで定義されている 1 つ以上の RSCT ピア・ドメイン上のグローバル・リソース・クラスに対してクラス・アクションを起動します。グローバル・クラスは、ドメインに関する情報が入っているリソース・クラスのピア・ドメインおよび管理ドメインで使用されます。管理サーバーで定義されているすべてのピア・ドメイン上のグローバル・リソース・クラスに対してクラス・アクションを起動するには、**CT\_MANAGEMENT\_SCOPE=3** を設定し、**-c** フラグを使用します。

**-f** *resource\_data\_input\_file*

リソースのアクション入力の要素と値が含まれているファイルの名前を指定します。**-i** フラグ指定で **lsactdef** コマンドを使用して、この入力ファイルのテンプレートを生成します。

**-d** 区切り文字により書式設定された出力を指定します。デフォルトの区切り文字はコロン (:) です。デフォルトの区切り文字を変更するには、**-D** フラグを使用します。

**-D** *delimiter*

指定した区切り文字により書式設定された出力を指定します。このフラグを使用して、デフォルトのコロン (:) 以外の区切り文字を指定します。1 つの例は、表示するデータにコロンが含まれている場合です。このフラグを使用して、1 つ以上の文字からなる区切り文字を指定します。

**-l** 長い形式、つまり 1 行に 1 項目の形式を指定します。これは、デフォルトの表示形式です。

## **-N** { *node\_file* | "-" }

ファイルまたは標準入力からノード名を読み取ることを指定します。ノード名がファイルにあることを示すには、**-N** *node\_file* を指定します。

- *node\_file* では 1 行に 1 つのノード名を指定します。
- 1 桁目の番号記号 (#) はその行がコメントであることを示します。
- ノード名の左のブランク文字はすべて無視されます。
- ノード名の右のブランク文字はすべて無視されます。

標準入力からノード名を読み取るには、**-N** "-" を使用します。

CT\_MANAGEMENT\_SCOPE 環境変数は、クラスターの有効範囲を決定します。

CT\_MANAGEMENT\_SCOPE が設定されていない場合、コマンドにとって有効な範囲になるまで、最初に管理ドメイン有効範囲 (存在する場合) を選択し、次にピア・ドメイン有効範囲 (存在する場合) を選択し、次にローカル有効範囲を選択します。コマンドは、最初に見つかった有効範囲に対して、1 回実行されます。例えば、管理ドメインとピア・ドメインの両方が存在する場合に CT\_MANAGEMENT\_SCOPE が設定されていないと、このコマンドは管理ドメインに適用されます。このコマンドをピア・ドメインに適用するには、CT\_MANAGEMENT\_SCOPE を **2** に設定します。

- t** 表形式を指定します。属性ごとにそれぞれ別個の列に表示され、1 行に 1 個のリソースが表示されます。
- x** ヘッダーを出力しないようにします。
- h** コマンドの使用ステートメント (使用法) を標準出力に書き込みます。
- T** コマンドのトレース・メッセージを標準エラーに書き込みます。ソフトウェア・サービスの編成の使用の場合のみです。
- V** コマンドの詳細メッセージを標準出力に書き込みます。

## パラメーター

*resource\_class* アクションを起動する対象となるリソース・クラスの名前を指定します。

*action* 起動するアクションの名前を指定します。

*in\_element=value...*

アクション入力要素の名前と値を指定します。 **-f** フラグを使用する場合、コマンド行に *in\_element=value* のペアを入力しないでください。

*in\_element* は、入力となる構造化データ要素名のいずれかです。指定されたアクションについて、定義されている構造化データ (SD) 入力要素ごとに、それぞれ 1 つの *in\_element\_n=value* ペアがなければなりません。特定のリソース・クラスおよびアクションの入力要素のリストを入手するには、**-s i** フラグを指定した **lsactdef** を使用します。**lsactdef -i** を使用して、入力ファイル・テンプレートを生成します。このテンプレートは、適切な編集処理を行って、入力ファイルとして使用できます。

*value* は、指定した要素に該当するデータ・タイプでなければなりません。例えば、**NodeNumber** が **uint32** データ・タイプとして定義されている場合は、正の数値を入力します。

*resource\_class* アクションを起動する対象となるリソース・クラスの名前を指定します。

### *resource\_handle*

アクションを起動する対象となるリソースおよびクラスのリソース・ハンドルを指定します。

### *rsp\_element*

1 つ以上のアクション応答構造化データ要素の名前を指定します。1 つ以上の要素名を指定すると、それらの要素だけが指定した順序で表示されます。要素名を指定しない場合は、応答のすべての要素が表示されます。

## セキュリティ

このコマンドには、**root** 権限が必要です。

## 終了状況

- 0 コマンドが正常に実行されました。
- 1 RMC でエラーが起きました。
- 2 コマンド行インターフェース (CLI) スクリプトでエラーが発生しました。
- 3 コマンド行に間違ったフラグが指定されました。
- 4 コマンド行に不正なパラメーターが指定されました。
- 5 不正なコマンド行入力に基づく RMC により、エラーが発生しました。

## 環境変数

### CT\_CONTACT

CT\_CONTACT 環境変数をホスト名または IP アドレスに設定すると、コマンドは指定されたホストの RMC (リソース・モニターおよび制御) デーモンと連絡します。この環境変数を指定しないと、コマンドは、そのコマンドが実行されているローカル・システムの RMC デーモンと連絡します。コマンドにより表示または変更されるリソースまたはリソース・クラスは、接続が確立されたシステム上で探されます。

### CT\_IP\_AUTHENT

CT\_IP\_AUTHENT 環境変数が存在する場合、RMC デーモンは、IP ベース・ネットワーク認証を使用して、CT\_CONTACT 環境変数が設定されている IP アドレスで指定されたシステム上の RMC デーモンに連絡します。CT\_IP\_AUTHENT は、CT\_CONTACT が IP アドレスに設定されている場合にのみ意味を持ちます。ドメイン・ネーム・システム (DNS) サービスには依存しません。

### CT\_MANAGEMENT\_SCOPE

RMC デーモンがリソースおよびリソース・クラスをモニターし、制御するためのセッションに使用する管理有効範囲を指定します。管理有効範囲は、リソースおよびリソース・クラスをモニターし、制御する対象となる、可能なターゲット・ノードのセットを決定します。有効な値は以下のとおりです。

- 0 *local* 有効範囲を指定します。
- 1 *local* 有効範囲を指定します。
- 2 *peer domain* (ピア・ドメイン) 有効範囲を指定します。
- 3 *management domain* (管理ドメイン) 有効範囲を指定します。

この環境変数を設定しないと、*local* 有効範囲が使用されます。

## 実装上の固有な条件

このコマンドは、AIX 用の Reliable Scalable Cluster Technology (RSCT) ファイルセットの一部です。

## 標準出力

**-h** フラグが指定されている場合は、このコマンドの使用状況の説明文が標準出力に書き込まれます。

コマンド出力とすべての詳細メッセージは、標準出力に書き込まれます。

## 標準エラー

トレース・メッセージはすべて、標準エラーに書き込まれます。

## 例

1. リソース・クラス **IBM.Foo** に対して **TestClassAction** リソース・クラス・アクションを起動するには、次のように入力します。

```
runact -c IBM.Foo TestClassAction Int32=99
```

出力は次のようになります。

```
Resource Class Action Response for: TestClassAction
sd_element 1:
  Int32 = 99
```

## 位置

**/usr/sbin/rsct/bin/runact**      **runact** コマンドが入っています。

## 関連情報

ブック: RMC 操作については、「*RSCT Administration Guide*」を参照してください。

コマンド: **lsactdef**

情報ファイル:

- **resource\_data\_input**
- **rmccli** (RMC 関連コマンドに関する一般情報)

---

## runcat コマンド

### 目的

**mkcatdefs** コマンドからの出力データをパイプ経由で **gencat** コマンドに渡します。

### 構文

```
runcat CatalogName SourceFile [ CatalogFile ]
```

### 説明

**runcat** コマンドは、**mkcatdefs** コマンドを呼び出して、メッセージ・カタログ・ソース・データ (**mkcatdefs** からの出力) を、**gencat** プログラムにパイプ接続します。



*SourceFile* パラメーターで指定されたファイルには、記号 ID 付きのメッセージ・テキストが入っています。**mkcatdefs** プログラムは、*CatalogName* パラメーターを使用し、**\_msg.h** を *CatalogName* 値の終わりに追加することによって、記号定義ファイルの名前を生成します。また、*CatalogName* 値の始めに **MF\_** を追加することによって、カタログ・ファイルのシンボル名を生成します。定義ファイルは、アプリケーション・プログラムに含めなければなりません。カタログ・ファイルのシンボル名は、ライブラリー関数 (例えば、**catopen** サブルーチン) の中で使用できます。

*CatalogFile* パラメーターは、**gencat** コマンドによって作成されたカタログ・ファイルの名前です。このパラメーターを指定しないと、**gencat** コマンドは、*CatalogName* 値の終わりに **.cat** を追加することによって、カタログ・ファイルに名前を付けます。このファイル名は、**catopen** ライブラリー関数の中で使用することもできます。

## 例

test.cat という名前のカタログを、メッセージ・ソース・ファイル test.msg から生成するには、下記のように入力します。

```
runcat test test.msg
```

## ファイル

**/usr/bin/runcat** **runcat** コマンドが入っています。

## 関連情報

**dspcat** コマンド、**dspmsg** コマンド、**gencat** コマンド、**mkcatdefs** コマンド。

**catclose** サブルーチン、**catgets** サブルーチン、**catopen** サブルーチン。

AIX 5L バージョン 5.3 ナショナル・ランゲージ・サポート ガイドおよびリファレンスのメッセージ機能のセクション。

---

## runlpcmd コマンド

### 目的

最小特権 (LP) リソースを実行します。

### 構文

LP リソースを実行するには、次のように入力します。

- ローカル・ノードの場合:

```
runlpcmd -N resource_name | RunCmdName [-h] [-TV] ["flags_and_parms"]
```

- ドメイン内のすべてのノードで以下のように入力します。

```
runlpcmd -a -N resource_name | RunCmdName [-h] [-TV] ["flags_and_parms"]
```

- ドメイン内の一部のノードで以下のように入力します。

```
runlpcmd -n host1 [,host2,...] -N resource_name | RunCmdName [-h] [-TV] ["flags_and_parms"]
```



## 説明

**runlpcmd** コマンドは、LP リソースを実行します。この LP リソースは、**root** コマンドまたはスクリプトであり、このコマンドまたはスクリプトに対して、LP のアクセス制御リスト (ACL) 内のアクセス権に基づいてユーザーはアクセス許可されます。**runlpcmd** コマンドを使用して、特定の *RunCmdName* 値に対応する LP コマンドを呼び出すことができます。この場合に呼び出し側ユーザーの権限と合致するアクセス権を指定します。**runlpcmd** が **-N** フラグ指定で呼び出されると、*resource\_name* パラメーターで指定された LP コマンドを実行します。*flags\_and\_parms* パラメーターを使用してコマンド呼び出しに必要なすべてのパラメーターとフラグを指定します。このパラメーターが無指定の場合、空ストリングが LP コマンドに渡されます。これはデフォルトです。

**Checksum** 属性値が **0** の場合、**ControlFlags** 値が **Checksum** の検査のために設定されていれば、**runlpcmd** はエラーを戻します。それ以外の場合は、エラーは戻されません。LP コマンドの **ControlFlag** 属性が、LP コマンドの実行前に **Checksum** を妥当性検査するのに設定された場合は、**runlpcmd** はそのような検査を行います。このコマンドが実行されるのは、計算された **Checksum** が対応する **Checksum** 属性の値と合致する場合だけです。この 2 つが合致していない場合は、コマンドはリジェクトされます。ただし、**ControlFlags** 属性がデフォルト値に設定されている場合は、**Checksum** の妥当性検査は行われません。

*RunCmdName* パラメーターを、**-N resource\_name** というフラグとパラメーターの組み合わせとともに指定することができます。ただし、*RunCmdName* パラメーターを使用する場合は、1 つの制限が適用されません。複数リソースが *RunCmdName* 値および呼び出し側のユーザーのアクセス権と合致する場合は、**runlpcmd** はエラーを戻します。*RunCmdName* 値および呼び出し側のユーザーのアクセス権に合致するリソースが 1 つある場合は、**runlpcmd** の *RunCmdName* は正常に戻ります。この制限を回避するために、**runlpcmd** により、ユーザーが **-N resource\_name** というフラグとパラメーターの組み合わせを使用して、固有の名前を指定することにより LP コマンドを実行することもできます。

LP コマンドを呼び出す前に、**runlpcmd** は、**FilterScript** 値が存在するかどうかを調べます。値が存在する場合は、このコマンドは、コマンド行に指定されている **FilterArg** 値と *flags\_and\_parms* パラメーター・ストリングを **FilterScript** に渡します。**FilterScript** が **0** を戻した場合は、**runlpcmd** は LP コマンドを呼び出します。**FilterScript** の実行によりゼロ以外の値が戻された場合は、**runlpcmd** はエラーを戻します。**FilterScript** が空であった場合は、**runlpcmd** は、**ControlFlags** で指定された一部の検査を行ってから、LP コマンドを直接呼び出します。

このコマンドの出力には、最終行として "**RC=return\_code**" が組み込まれることがあります。

このコマンドはどのノードでも実行できます。ドメイン内のすべてのノードでこのコマンドを実行したい場合には、**-a** フラグを使用します。ドメイン内の一部のノードでこのコマンドを実行したい場合には、**-n** フラグを使用します。それ以外の場合、このコマンドはローカル・ノードで実行されます。

## フラグ

- a**     ドメイン内のすべてのノードの 1 つ以上のリソースを変更します。**CT\_MANAGEMENT\_SCOPE** 環境変数の設定値によって、クラスター有効範囲が決まります。**CT\_MANAGEMENT\_SCOPE** 変数が設定されていない場合、LP リソース・マネージャーは次の順序で有効範囲設定値を使用します。
  1. 管理ドメイン (あれば)
  2. ピア・ドメイン (あれば)
  3. ローカル有効範囲

**runlpcmd** コマンドは、LP リソース・マネージャーが最初に検出した有効範囲に対して 1 回実行されます。例えば、管理ドメインとピア・ドメインが存在していて、かつ、その **CT\_MANAGEMENT\_SCOPE** 環境変数が設定されていないと仮定します。この場合、**runlpcmd -a** は管理ドメインで実行されます。**runlpcmd -a** をピア・ドメインで実行するには、**CT\_MANAGEMENT\_SCOPE** を **2** に設定する必要があります。

**-n** *host1[,host2,...]*

LP リソースを変更するドメインのノード (複数の場合もある) を指定します。デフォルトでは、LP リソースは、ローカル・ノードで変更されます。**-n** フラグが有効であるのは、管理ドメインまたはピア・ドメインの中だけです。**CT\_MANAGEMENT\_SCOPE** 変数が設定されていない場合、LP リソース・マネージャーは次の順序で有効範囲設定値を使用します。

1. 管理ドメイン (あれば)
2. ピア・ドメイン (あれば)
3. ローカル有効範囲

**runlpcmd** コマンドは、LP リソース・マネージャーが最初に検出した有効範囲に対して 1 回実行されます。

**-N** *resource\_name*

ドメイン内の 1 つ以上のノードで実行したい LP リソースの名前を指定します。

**-h** コマンドの使用ステートメント (使用法) を標準出力に書き込みます。

**-T** コマンドのトレース・メッセージを標準エラーに書き込みます。

**-v** コマンドの詳細メッセージを標準出力に書き込みます。

## パラメーター

*RunCmdName*

ドメイン内の 1 つ以上のノードで実行したい LP リソースの名前を指定します。

*"flags\_and\_parms"*

LP コマンドまたはスクリプトの入力として必要な、フラグおよびパラメーターを指定します。このパラメーターを指定しないと、空ストリングが LP コマンドに渡されます。これはデフォルトです。

## セキュリティ

**runlpcmd** コマンドを実行するには、以下のアクセス権が必要です。

- **IBM.LPCCommands** リソース・クラスの Class ACL の中の読み取り許可。
- Resource ACL の中の実行アクセス権。

代わりに、この許可が Resource Shared ACL の中に存在する場合は、Resource ACL が Resource Shared ACL の使用を指示できます。

アクセス権は、連絡先システムの LP ACL で指定されています。LP ACL に関する一般情報については **lpaci** 情報ファイル、LP ACL の変更については「*RSCT: Administration Guide*」を参照してください。

## 終了状況

- 0 コマンドが正常に実行されました。
- 1 RMC でエラーが起きました。
- 2 コマンド行インターフェース (CLI) スクリプトでエラーが発生しました。

- 3 コマンド行に間違っただフラグが指定されました。
- 4 コマンド行に不正なパラメーターが指定されました。
- 5 不正なコマンド行入力に基づく RMC により、エラーが発生しました。
- 6 リソースが見つかりません。

## 環境変数

### CT\_CONTACT

RMC デーモンとのセッションに使用されるシステムを決めます。 **CT\_CONTACT** にホスト名または IP アドレスが設定されていると、このコマンドは指定されたホスト上の RMC デーモンにアクセスします。この環境変数を指定しないと、コマンドは、そのコマンドが実行されているローカル・システムの RMC デーモンと連絡します。 RMC デーモン・セッションのターゲットおよびその管理有効範囲によって、処理される LP リソースが決まります。

### CT\_MANAGEMENT\_SCOPE

LP リソースを処理するために RMC デーモンとのセッションに使用される管理有効範囲を決定します。管理有効範囲は、リソースを処理できる、可能なターゲット・ノードのセットを決定します。有効な値は以下のとおりです。

- 0 *local* 有効範囲を指定します。
- 1 *local* 有効範囲を指定します。
- 2 *peer domain* (ピア・ドメイン) 有効範囲を指定します。
- 3 *management domain* (管理ドメイン) 有効範囲を指定します。

この環境変数を設定しないと、ローカルの 有効範囲が使用されます。

## 実装上の固有な条件

このコマンドは、AIX 用の Reliable Scalable Cluster Technology (RSCT) ファイルセットの一部です。

## 標準出力

**-h** フラグが指定されている場合は、このコマンドの使用状況の説明文が標準出力に書き込まれます。 **-v** フラグが指定されると、このコマンドの詳細メッセージが標準出力に書き込まれます。

## 標準エラー

トレース・メッセージはすべて、標準エラーに書き込まれます。

## 例

**LP1** という名前の LP リソースを実行するには、フラグとパラメーター **-a -p User Group** が必要で、次のように入力します。

```
runlpcmd LP1 "-a -p User Group"
```

## 位置

**/usr/sbin/rsct/bin/runlpcmd** **runlpcmd** コマンドが入っています。

## 関連情報

ブック: LP ACL の変更については、「*RSCT Administration Guide*」を参照してください。

コマンド: **chlpcmd**、**lphistory**、**lslpcmd**、**mklpcmd**、**rmlpcmd**

情報ファイル: **lpacl** (LP ACL に関する一般情報)

---

## rup コマンド

### 目的

ローカル・ネットワーク上のリモート・ホストの状況を表示します。

### 構文

```
/usr/bin/rup [ -h | -l | -t ] [ Host ... ]
```

### 説明

**rup** コマンドは、ローカル・ネットワーク上でブロードキャストを行い、受信した応答を表示することにより、リモート・ホストの状況を表示します。出力結果をソートしたい場合には、フラグを指定してください。フラグを指定しないと、**rup** コマンドは応答を受信した順序どおりに表示します。コマンド行で複数のホストを指定すると、**rup** コマンドはフラグをすべて無視し、指定されたホストの順序どおりに出力結果を表示します。出力結果をソートする場合には、必ず **sort** コマンドを使用してください。

さらに、*Host* パラメーターに値を指定すると、**rup** コマンドは、全ホストにブロードキャストを行うのではなく、指定されたホストを照会します。リモート・ホストが応答するのは、**rstatd** デーモンを実行している場合に限りです。このデーモンは通常、**inetd** デーモンから開始されます。

#### 注:

- ブロードキャストは、ゲートウェイを経由すると機能しません。したがって、ホストを指定しない場合には、使用中のネットワーク上のホストだけが、**rup** コマンドに応答可能です。
- ロード平均統計情報は、カーネルにより保持されません。ロード平均は、このコマンドにより、常に 0 (ゼロ) として報告されます。

### フラグ

- h** 表示内容を、アルファベット順にホスト名でソートします。
- l** 表示内容をロード平均でソートします。
- t** 表示内容を、ネットワーク上の実行時間の長さでソートします。

### 例

- ネットワーク上の全ホストの状況を検出し、そのリストをアルファベット順にホスト名でソートするには、次のように入力します。

```
/usr/bin/rup -h
```

- ネットワーク上の全ホストのリストを、各コンピューターのロード平均を基準にして表示するには、次のように入力します。

```
/usr/bin/rup -l
```

- ホストの状況を表示するには、次のように入力します。

```
/usr/bin/rup brutus
```

この例では、**rup** コマンドにより、**brutus** という名前のホストの状況が表示されます。

- ネットワーク上の全ホストの状況を、各コンピューターの実行時間の長さでソートして表示するには、次のように入力します。

```
/usr/bin/rup -t
```

## ファイル

**/etc/inetd.conf**                    **inetd** デーモンがインターネット・サービス要求を処理する方法を定義します。

## 関連情報

ネットワークおよびコミュニケーションのマネージ の **sort** コマンド、NFS コマンドのリストのセクション。

**inetd** デーモン、 **rstatd** デーモン。

ネットワークおよびコミュニケーションのマネージ のネットワーク・ファイルシステム (NFS) の概要のセクション。

---

## ruptime コマンド

### 目的

ネットワーク上の各ホストの状況を表示します。

### 構文

```
ruptime [ -a ] [ -r ] [ -l | -t | -u ]
```

### 説明

**/usr/bin/ruptime** コマンドは、ローカル・ネットワーク上で **rwhod** デーモンを実行している各ホストの状況を表示します。状況表示行は、**-l**、**-t**、または **-u** フラグを指定しない限り、ホスト名でソートされます。状況情報は、**rwhod** デーモンを実行している各ネットワーク・ホストによって、3 分に 1 回の割合で、パケット・ブロードキャストで提供されます。ブロードキャスト間に発生したアクティビティ（例えば、ホストの電源のオン、オフなど）は、次のブロードキャストまで反映されません。11 分間何も状況情報が受信されないホストは、ダウンとして報告されます。

出力内容は、ホスト名、状況、時刻、ユーザー数、ロード平均というフォーマットです。ロード平均には、サーバーの送信前の、1 分、5 分、15 分のインターバルでのロード平均が表示されます。ロード平均には 10 が乗算されて、10 進数フォーマットの値が表示されます。

## フラグ

- a** すべてのユーザーを含みます。このフラグを指定しないと、セッションが 1 時間以上アイドルの状態にあるユーザーが含まれません。
- l** リストをロード平均でソートします。
- r** ソートの順序を逆にします。**-r** フラグは、**-l**、**-t** または **-u** フラグと一緒に使用しなければなりません。
- t** リストをアップタイムでソートします。
- u** リストをユーザー数でソートします。

## 例

1. ローカル・ネットワーク上のホストの状況レポートを取得するには、次のように入力します。

```
ruptime
```

以下のような情報が表示されます。

```
host1    up      5:15,  4 users,  load 0.09, 0.04, 0.04
host2    up      7:45,  3 users,  load 0.08, 0.07, 0.04
host7    up      7:43,  1 user,   load 0.06, 0.12, 0.11
```

2. ロード平均でソートされた状況レポートを入手するには、下記のように入力します。

```
ruptime -l
```

以下のような情報が表示されます。

```
host2    up      7:45,  3 users,  load 0.08, 0.07, 0.04
host1    up      5:18,  4 users,  load 0.07, 0.07, 0.04
host7    up      7:43,  1 user,   load 0.06, 0.12, 0.11
```

## ファイル

`/var/spool/rwho/whod.*`

リモート **rwhod** デーモンから受け取ったデータ・ファイルを示します。

## 関連情報

**rwho** コマンド。

**rwhod** デーモン。

ネットワークおよびコミュニケーションのマネージの通信およびネットワークのセクション。

---

## ruser コマンド

### 目的

外部ホストがプログラムにアクセスするのを制御する 3 つの別々のシステム・データベース内のエントリーを直接操作します。

### 構文

#### データベース・ファイルの名前エントリーを追加または削除する

```
ruser { -a | -d } { -f "UserName ..." | -p "HostName ..." | -r "HostName ..." }
```



## データベース・ファイル内のすべての名前エントリーを削除または表示する

```
ruser { -X | -s } { -F | -P | -R } [ -Z ]
```

### 説明

**ruser** 低水準コマンドは、3 つの独立したシステム・データベースにエントリーを追加または削除します。操作の対象となるデータベースは、**-p** フラグ、**-r** フラグ、または **-f** フラグを使用することによって決定されます。また、**ruser** コマンドは、データベースのうちの 1 つに含まれる 1 つまたはすべてのエントリーを表示できます。各データベースは、名前のリストになっています。3 つのデータベースは下記のとおりです。

- **/etc/ftpusers** ファイル
- **/etc/hosts.equiv** ファイル
- **/etc/hosts.lpd** ファイル

**注:** **-p** オプションと **-r** オプションを組み合わせることで、それぞれのデータベースに同時に名前を追加できますが、**-f** オプションはいずれのオプションとも組み合わせることはできません。

Web-based System Manager のユーザー・アプリケーションを使用して、ユーザー特性を変更することができます。

このコマンドを実行するには、System Management Interface Tool (SMIT) **smit users** 高速パスを使用するか、または次のように入力することもできます。

```
smit rprint
```

### フラグ

- a** 名前をデータベースに追加します。 **-a** フラグは、 **-p** フラグ、 **-r** フラグ、または **-f** フラグのいずれかと一緒に使用しなければなりません。
- d** 名前をデータベースから削除します。このフラグは、 **-p** フラグ、 **-r** フラグ、または **-f** フラグのいずれかと一緒に使用しなければなりません。
- F** **/etc/ftpusers** ファイル内のすべてのエントリーを削除または表示します。このフラグと **-X** フラグを組み合わせることで、すべてのエントリーを削除します。このフラグと **-s** フラグを組み合わせることで、すべてのエントリーを表示します。
- f "UserName ..."** リモート FTP クライアントが使用できないローカル・ユーザー名のリストを含む **/etc/ftpusers** データベースに対して、 **UserName** 変数によって指定されたユーザー名の追加または削除を行います。 **-f** フラグは、 **-a** フラグまたは **-d** フラグのいずれかと一緒に使用しなければなりません。
- P** **/etc/hosts.lpd** ファイル内のすべてのエントリーを削除または表示します。このフラグと **-X** フラグを組み合わせることで、すべてのエントリーを削除します。このフラグと **-s** フラグを組み合わせることで、すべてのエントリーを表示します。
- p "HostName ..."** どの外部ホストがユーザーのコンピューター上で出力できるかどうかを指定するデータベース内で、 **HostName** 変数で指定されたホスト名の追加または削除を行います。 **-p** フラグは、 **-a** フラグまたは **-d** フラグのいずれかと一緒に使用しなければなりません。
- R** **/etc/hosts.equiv** ファイル内のすべてのエントリーを削除または表示します。このフラグと **-X** フラグを組み合わせることで、すべてのエントリーを削除します。このフラグと **-s** フラグを組み合わせることで、すべてのエントリーを表示します。



|                          |                                                                                                                                                                                                                                                              |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-r</b> "HostName ..." | どの外部ホストがユーザーのコンピューター上でリモート・コマンド ( <b>rlogin</b> 、 <b>rnp</b> 、 <b>rsh</b> 、または <b>print</b> ) を実行できるかを指定する <b>/etc/hosts.equiv</b> データベース内で、 <i>HostName</i> 変数で指定されたホスト名の追加または削除を行います。 <b>-r</b> フラグは、 <b>-a</b> フラグまたは <b>-d</b> フラグのいずれかと一緒に使用しなければなりません。 |
| <b>-s</b>                | データベース内のすべてのエントリーを表示します。このフラグは、 <b>-P</b> フラグ、 <b>-R</b> フラグ、または <b>-F</b> フラグのいずれかと一緒に使用しなければなりません。                                                                                                                                                         |
| <b>-X</b>                | すべての名前をデータベースから削除します。このフラグは、 <b>-P</b> フラグ、 <b>-R</b> フラグ、または <b>-F</b> フラグのいずれかと一緒に使用しなければなりません。                                                                                                                                                            |
| <b>-Z</b>                | <b>-s</b> フラグは、 <b>-Z</b> フラグを指定した場合に必要です。 <b>-Z</b> フラグを指定した場合、データベースの表示の前に簡潔なタイトルが表示されます。                                                                                                                                                                  |

## 例

1. どの外部ホストがローカル・コンピューター上で印刷できるかを指定する **/etc/hosts.lpd** データベースにエントリーを追加するには、コマンドを下記のフォーマットで入力します。

```
ruser -a -p "host1"
```

この例では、外部ホストは `host1` です。

2. 印刷のみを制御するデータベース (**/etc/hosts.lpd**) のエントリーを削除し、さらに **rlogin** コマンド、**rnp** コマンド、および **rsh** コマンドのリモート・アクセスを制御するデータベース (**/etc/hosts.equiv**) から、同じ名前を削除するには、次のように入力します。

```
ruser -d -r "host2" -p "host1"
```

この例で、データベース・エントリーが削除されるホストは `host1` です。

## 関連情報

**lpd** デーモン、**rshd** デーモン、**rlogind** デーモン。

**ftputers** ファイル・フォーマット、**hosts.equiv** ファイル・フォーマット、**hosts.lpd** ファイル・フォーマット。

Web-based System Manager のインストールについては、*AIX 5L Version 5.3 Web-based System Manager 管理ガイド*の第 2 章 インストールおよびシステム要件のセクションを参照してください。

---

## rusers コマンド

### 目的

リモート・マシンにログオンしたユーザーのリストを報告します。

### 構文

```
/usr/bin/rusers [ -a ] [ -l ] [ -u | -h | -i ] [ Host ...]
```

### 説明

**rusers** コマンドは、リモート・コンピューターにログオンしたユーザーのリストを作成します。**rusers** コマンドは、そのために、ローカル・ネットワーク上の各コンピューターに対してブロードキャストを行い、受信した応答を出力することにより、リストを作成します。通常、システムは、受信した順序で応答を

出力します。この順序を変更するには、いずれかのフラグを指定します。さらに、*Host* パラメーターを指定すると、**rusers** コマンドは、全ホストにブロードキャストを行うのではなく、指定された 1 台以上のホストを照会します。

デフォルトでは、各エントリーに、各コンピューターのユーザーのリストが含まれます。これらのエントリーには、そのコンピューターにログインしている全ユーザーの名前が含まれます。さらに、ユーザーが 1 分間以上システムに入力を行わないと、**rusers** コマンドは、そのユーザーのアイドル時間を報告します。

リモート・ホストが応答するのは、**rusersd** デーモンを実行している場合に限りです。このデーモンは通常、**inetd** デーモンから開始されます。

**注:** ブロードキャストは、ゲートウェイを経由すると機能しません。したがって、ホストを指定しない場合には、使用中のネットワーク上のホストだけが、**rusers** コマンドに応答可能です。

## フラグ

- a ログインしているユーザーが存在しない場合でも、コンピューターのレポートを作成します。
- h アルファベット順にホスト名でソートします。
- i アイドル時間でソートします。
- l **who** コマンドと似た長めのリストを作成します。
- u ユーザー数でソートします。

## 例

1. リモート・マシンにログインしているネットワーク上のユーザーのリストを作成するには、次のように入力します。

```
rusers
```

2. ユーザーのリストを、アルファベット順にホスト名でソートした形で作成するには、次のように入力します。

```
rusers -h
```

3. ホスト上のユーザーのリストを作成するには、次のように入力します。

```
rusers -h pluto
```

この例では、**rusers** コマンドにより、*pluto* という名前のホスト上のユーザーのリストが作成されます。

4. リモート・マシンにログインしているユーザーのリストを、各コンピューターのアイドル時間の長さでソートした形で作成するには、次のように入力します。

```
rusers -i
```

5. リモート・マシンにログインしているユーザーのリストを、ログイン・ユーザー数でソートした形で作成するには、次のように入力します。

```
rusers -u
```

## ファイル

**/etc/inetd.conf**

RPC デーモンおよび他の TCP/IP デーモンを開始する TCP/IP 構成ファイル。

## 関連情報

**rwho** コマンド、 **who** コマンド。

**inetd** デーモン、 **rusersd** デーモン。

ネットワークおよびコミュニケーションのマネージ のネットワーク・ファイルシステム (NFS) の概要のセクション。

ネットワークおよびコミュニケーションのマネージ の NFS コマンドのリストのセクション。

---

## rusersd デーモン

### 目的

**rusers** コマンドからの照会に応答します。

### 構文

**/usr/lib/netsvc/rusers/rpc.rusersd**

### 説明

**rusersd** デーモンは、ネットワーク上のユーザーの現行リストを戻すことによって、**rusers** コマンドからの照会に응答するサーバーです。通常、このデーモンは、**inetd** デーモンによって始動されます。

### ファイル

|                        |                                               |
|------------------------|-----------------------------------------------|
| <b>/etc/inetd.conf</b> | RPC デーモンおよび他の TCP/IP デーモンを開始する TCP/IP 構成ファイル。 |
| <b>/etc/utmp</b>       | システムにログインされたユーザーに関する情報が入っています。                |

## 関連情報

**rusers** コマンド。

**inetd** デーモン。

ネットワークおよびコミュニケーションのマネージ のネットワーク・ファイルシステム (NFS) の概要のセクション。

ネットワークおよびコミュニケーションのマネージ の NFS コマンドのリストのセクション。

---

## rvsdrestrict コマンド

### 目的

**rvsdrestrict** - リカバリー可能仮想共用ディスク・サブシステムの実行レベルを表示し、設定します。このコマンドは、RVSD サブシステムの開始前に発行しなければなりません。

### 構文

**rvsdrestrict** {-l | -s {RVSD4.1 | RESET}}

## 説明

**rvsdrestrict** コマンドは、リカバリー可能仮想共用ディスクサブシステムが実行されるレベルを制限するために使用します。このコマンドを使用して設定したレベルよりも、インストール済みの RVSD ソフトウェアのノードのレベルが低い場合は、RVSD サブシステムはそのノードで開始されません。

このコマンドは、ピア・ドメイン全体に渡る RVSD サブシステムの実行レベルを動的には変更しません。この情報に対しては、再始動後に 1 つの RVSD サブシステムのインスタンスだけが反応します。ご使用のピア・ドメインがあるレベルで実行されている場合に、そのレベルをオーバーライドするには、以下の手順を行います。

1. すべてのノードで RVSD サブシステム停止します。
2. レベルをオーバーライドします。
3. RVSD サブシステムを再始動します。

## フラグ

- l 現行の RVSD サブシステム実行レベルをリストします。
- s RVSD サブシステム実行レベルを設定します。

## パラメーター

なし。

## セキュリティ

このコマンドを実行するには、root 権限を持っている必要があります。

## 終了状況

- 0 コマンドが正常終了したことを示します。
- nonzero エラーが発生したことを示します。

## 制約事項

このコマンドはピア・ドメイン内のオンラインのノードから発行する必要があります。オンラインのピア・ドメインに移動するには **starttrpdomain** コマンドを使用します。既存のピア・ドメイン内で、オンラインの特定ノードに移動するには **starttrpnode** コマンドを使用します。RSCT ピア・ドメインの作成と管理の詳細については、「*RSCT: Administration Guide*」を参照してください。

## 標準出力

現行の RVSD サブシステム実行レベル。

## 例

1. RVSD サブシステム実行レベルを RVSD4.1 に設定するには、次のコマンドを発行します。

```
rvsdrestrict -s RVSD4.1
```

## 位置

**/opt/rsct/vsd/bin/rvsdrestrict**

## 関連情報

コマンド: `cfgvsd`、`lsvsd`、`preparevsd`、`startvsd`、`stopvsd`、`suspendvsd`、`ucfgvsd`

---

## rwall コマンド

### 目的

ネットワーク上の全ユーザーにメッセージを送信します。

### 構文

#### 指定したホストにメッセージを送信する

```
/usr/sbin/rwall HostName ...
```

#### 指定したネットワークにメッセージを送信する

```
/usr/sbin/rwall -n NetworkGroup ...
```

#### ネットワーク上の指定されたホストにメッセージを送信する

```
/usr/sbin/rwall -h HostName ... -n NetworkGroup
```

### 説明

**rwall** コマンドは、ネットワーク上の全ユーザーにメッセージを送信します。メッセージを送信するために、**rwall** コマンドは、ファイル終わり文字を検出するまで、標準入力からメッセージを読み取り続けます。**rwall** コマンドは、Broadcast Message... という行で始まるこのメッセージを取り込み、指定のホスト・マシンにログインしている全ユーザーに、これをブロードキャストします。ユーザーがメッセージを受信するのは、**rwalld** デーモンを実行している場合に限りです。このデーモンは、**inetd** デーモンから開始されます。

**注:** タイムアウトはかなり短くなっています。これにより、**rwall** コマンドは、大きなコンピューターのグループ (一部のコンピューターがダウンしている可能性のあるグループ) に、合理的な時間内にメッセージを送信することができます。したがって、ロード過多のコンピューターには、メッセージが送信されないこともあります。

### フラグ

- h** *HostName* パラメーターに指定されたコンピューターにメッセージを送信します。
- n** 特定のネットワーク・グループのみにメッセージを送信します。ネットワーク・グループは、**netgroup** ファイルに定義されています。

### 例

1. `neptune` という名前のホストにメッセージを送信するには、以下のように入力します。

```
/usr/sbin/rwall neptune
```

メッセージをキー入力してください。メッセージの入力が終わったら、以下のように入力します。

```
Ctrl D
```

2. メッセージを `neptune` という名前のホスト、および `cosmos` ネットグループ内の各ホストに送信するには、下記のように入力します。

```
rwall -n cosmos -h neptune
```

メッセージをキー入力してください。メッセージの入力が終わったら、以下のように入力します。

```
Ctrl D
```

## ファイル

`/etc/inetd.conf`                   RPC デーモンおよび他の TCP/IP デーモンを開始する TCP/IP 構成ファイル。  
`/etc/netgroup`                   ネットワーク上の各ユーザー・グループに関する情報が入っています。

## 関連情報

`wall` コマンド。

`inetd` デーモン、`rwalld` デーモン。

ネットワークおよびコミュニケーションのマネージ のネットワーク・ファイルシステム (NFS) の概要のセクション。

ネットワークおよびコミュニケーションのマネージ の NFS コマンドのリストのセクション。

---

## rwalld デーモン

### 目的

`rwall` コマンドからの要求を処理します。

### 構文

```
/usr/lib/netsvc/rwall/rpc.rwalld
```

### 説明

`rwalld` デーモンは、`rwall` コマンドからの要求を処理します。`inetd` デーモンが `rwalld` デーモンを呼び出します。

## ファイル

`/etc/inetd.conf`                   TCP/IP 構成を指定します。

## 関連情報

`rwall` コマンド。

`inetd` デーモン。

ネットワークおよびコミュニケーションのマネージ のネットワーク・ファイルシステム (NFS) の概要のセクション。

---

## rwwho コマンド

### 目的

ローカル・ネットワーク上のホストにログインしているユーザーを表示します。

### 構文

**rwwho** [ **-a** ]

### 説明

**/usr/bin/rwho** コマンドは、**rwwho** デーモンを実行中のホストに現在ログインしているローカル・ネットワーク上のユーザーの、ユーザー名、ホスト名、各セッションの開始日時を表示します。ワークステーションが 3 分間以上非アクティブの状態にあると、**rwwho** コマンドは、最後のカラムにそのアイドル時間を分単位で報告します。非アクティブの状態が 1 時間を過ぎると、そのユーザーは、**-a** フラグが指定されない限り現在ログインしているユーザーに含まれなくなります。

**注:** このコマンドは多くの出力結果を表示するので、ローカル・ネットワーク上にかなりの数のユーザーが存在している場合にこのコマンドを使用するときには、嚴重な注意が必要です。

状況情報は、**rwwho** デーモンを実行中の各ネットワーク・ホストにより、3 分に一度ずつブロードキャストされます。ブロードキャスト間に発生するアクティビティー (ユーザーのログオン、ログオフなど) は、いずれも次のブロードキャストまで反映されません。

### フラグ

**-a** すべてのユーザーを含みます。このフラグを指定しないと、セッションが 1 時間以上アイドルの状態にあるユーザーが、レポートに含まれません。

### 例

ローカル・ネットワーク上のホストに現在ログインしている全ユーザーのレポートを取得するには、次のように入力します。

```
rwwho
```

以下のような情報が表示されます。

```
bob    host2:pts5    Nov 17 06:30 :20
bob    host7:console Nov 17 06:25 :25
fran   host1:pts0    Nov 17 11:20 :51
fran   host1:pts8    Nov 16 15:33 :42
fran   host4:console Nov 17 16:32
server host2:console Nov 17 06:58 :20
alice  host2:pts6    Nov 17 09:22
```

### ファイル

**/var/spool/rwho/whod.\***

リモート **rwwho** デーモンから受け取ったデータ・ファイルを示します。



## 関連情報

**ruptime** コマンド、 **rwho** コマンド、 **who** コマンド。

**gethostname** サブルーチン。

**services** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ の通信およびネットワークのセクション。

---

## rwhod デーモン

### 目的

**rwho** コマンドと **ruptime** コマンドのためのサーバー機能を提供します。

### 構文

注: コマンド行から **rwhod** デーモンを制御するには、**SRC** コマンドを使用します。システム始動時に毎回デーモンを始動するには、**rc.tcpip** ファイルを使用します。

**/usr/sbin/rwhod**

### 説明

**/usr/sbin/rwhod** デーモンは、**rwho** コマンドと **ruptime** コマンドによって使用されるデータベースを維持します。**rwhod** デーモンは、始動されると、状況情報の作成と利用の両方を行います。

状況情報を作成する際、**rwhod** デーモンは、約 3 分ごとに、ローカル・ホストの状態を照会します。次に、状況メッセージを構成し、これらをローカル・ネットワークにブロードキャストします。

状況情報を利用する際、**rwhod** デーモンは、リモート・ホスト上の **rwhod** サーバーからの状況メッセージを待ちます。**rwhod** デーモンは状況メッセージを受信すると、そのメッセージの妥当性を検査します。その次に、このメッセージを **/var/spool/rwho** ディレクトリーに記録します。( **rwho** コマンドと **ruptime** コマンドは、**/var/spool/rwho** ディレクトリー内のファイルを使用して、これらのコマンドの状況リストを生成します。)

**rwhod** デーモンは、**/etc/services** ファイルに指定されたとおりに **rwho** ソケットを使用して、状況メッセージのブロードキャストと受信を行います。

これらのメッセージを作成するとき、**rwhod** デーモンは、直前の 1 分、5 分、15 分のインターバルで、平均 CPU ロードのエントリーを計算します。これらのメッセージをブロードキャストする前に、**rwhod** デーモンは、ネットワークで使用可能なバイト・オーダーにこれらを変換します。

**rwhod** デーモンは、**rwho** ソケットでメッセージを受信すると、**rwho** ソケット以外から発信されたメッセージをすべて破棄します。また、印刷不能な ASCII 文字が含まれているメッセージもすべて破棄します。**rwhod** デーモンは、有効なメッセージを受信すると、それを **/var/spool/rwho** ディレクトリー内の **rwhod.HostName** ファイルに入れて、同じ名前を持つファイルを上書きします。

**rwhod** デーモンは、システム・リソース・コントローラー (SRC) を使用して制御する必要があります。コマンド行に **rwhod** を入力することはお勧めできません。

## システム・リソース・コントローラーを使用した rwhod デーモンの操作

**rwhod** デーモンは、システム・リソース・コントローラー (SRC) により制御されるサブシステムです。**rwhod** デーモンは **tcPIP** システム・グループのメンバーです。このデーモンはデフォルトでは使用不可で、以下の SRC コマンドにより操作できます。

|                  |                                              |
|------------------|----------------------------------------------|
| <b>stopsrc</b>   | サブシステム、サブシステムのグループ、あるいはサブサーバーを停止します。         |
| <b>traceson</b>  | サブシステム、サブシステムのグループ、あるいはサブサーバーのトレースを使用可能にします。 |
| <b>tracesoff</b> | サブシステム、サブシステムのグループ、あるいはサブサーバーのトレースを使用不可にします。 |
| <b>lssrc</b>     | サブシステム、サブシステムのグループ、あるいはサブサーバーのトレースの状況を取得します。 |

### 例

1. **rwhod** デーモンを始動するには、次のように入力します。

```
startsrc -s rwhod
```

このコマンドは、デーモンを始動します。このコマンドは、**rc.tcPIP** ファイル内、またはコマンド行上で使用できます。 **-s** フラグは、後に続くサブシステムを始動することを指定します。

2. **rwhod** デーモンを正常に停止するには、以下のように入力します。

```
stopsrc -s rwhod
```

このコマンドはデーモンを停止します。 **-s** フラグは、後に続くサブシステムを停止することを指定します。

3. **rwhod** デーモンに関する簡潔なフォーマットの状況レポートを入手するには、以下のように入力します。

```
lssrc -s rwhod
```

このコマンドにより、デーモン名、デーモンのプロセス ID、デーモンの状態 (アクティブか、非アクティブか) が戻されます。

4. **rwhod** デーモンのトレースを可能にするには、次のように入力します。

```
traceson -s rwhod
```

このコマンドにより、ソケット・レベルのデバッグが可能になります。このコマンド例の出力結果を見場合は、**trpt** コマンドを使用してください。

### ファイル

|                                      |                                                                               |
|--------------------------------------|-------------------------------------------------------------------------------|
| <b>/etc/utmp</b>                     | ローカル・ホストにログインしているユーザーの状況情報が入っています。                                            |
| <b>/var/spool/rwho/*</b>             | <b>rwho</b> コマンドと <b>ruptime</b> コマンドが、これらのコマンドの状況リストを生成するときに使用するファイルが入っています。 |
| <b>/var/spool/rwho/whod.HostName</b> | <b>HostName</b> パラメーターに指定されたホストの最新の状況情報が入っています。                               |

### 関連情報

**ruptime** コマンド、 **rwho** コマンド、 **who** コマンド。

**gethostname** サブルーチン。

**services** ファイル・フォーマット。

ネットワークおよびコミュニケーションのマネージ の TCP/IP デーモンのセクション。



## 付録. 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、製造元によって明示的に指定されたものを除き、他社の製品、プログラムまたはサービスを使用した場合の評価と検証はお客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒242-8502

神奈川県大和市下鶴間1623番14号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

**以下の保証は、国または地域の法律に沿わない場合は、適用されません。** IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation

Dept. LRAS/Bldg. 003

11400 Burnet Road

Austin, TX 78758-3498

U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

#### 著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。お客様は、IBM のアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれらのサンプル・プログラムを複製し、改変し、配布することができます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

---

## 商標

IBM、IBM ロゴおよび [ibm.com](http://ibm.com) は、世界の多くの国で登録された International Business Machines Corp. の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

Adobe、Adobe ロゴ、PostScript、PostScript ロゴは、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

UNIX は、The Open Group の米国およびその他の国における登録商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

# 索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

## [ア行]

- アカウントティング・システム
  - 基本ファイルの作成
    - nulladm コマンドを使用する 343
  - 合計アカウント・レコードのフォーマット
    - prtacct コマンドを使用する 556
- 実行
  - runacct コマンドを使用する 902
  - セッション・レコードの表示 507
  - 前日の ASCII レポートのフォーマット
    - prdaily コマンドを使用する 508
  - ファイルの削除
    - remove コマンドを使用する 716
  - プリンター・レコード
    - 作成 367
- イベント応答リソース・マネージャー (ERRM)
  - コマンド
    - notifyevent 313
  - スクリプト
    - notifyevent 313
- 印刷
  - ジョブの保持 618
  - スプールされたジョブの移動 619
- 印刷キュー
  - 新規の作成 448
  - 内のジョブの優先順位付け
    - qpri コマンドを使用する 627
- 印刷ジョブ・マネージャー
  - 印刷ジョブの処理
    - piobe コマンドを使用する 430
  - ファイルの印刷
    - pioout コマンドを使用する 452
- インストール可能パッケージ
  - 作成
    - pkgmk コマンドを使用する 465
- インターネット
  - ドメイン・ネーム・サーバーの照会 318
- エコー要求
  - ネットワーク・ホストに対する送信
    - ping コマンドを使用する 424
- オブジェクト・ファイル
  - 記号テーブルの表示 238
  - プロファイル・データの表示 534

## [カ行]

- 外部ホスト・アクセス
  - データベース内のエントリーの操作
    - ruser コマンドを使用する 916
- 獲得
  - ネットワーク状況
    - netstat コマンドを使用する 33
- 仮想 RAM ディスク
  - 除去
    - rmramdisk コマンドを使用する 830
- 仮想プリンター 435, 442
  - 除去
    - rmvirprt コマンドを使用する 861
    - の属性値の表示
      - lsvirprt コマンドを使用する 428
- 仮想メモリー
  - システムのページ・サイズの表示 374
- 環境
  - 変数値の印刷 515
- 疑似デバイス・プリンター 445
- キュー
  - スプールされたジョブの移動 619
  - スプールされたジョブの保持 618
- 共用ログイン・ポート 588
- グループ
  - 除去
    - rmgroup コマンドを使用する 804
- グループ ID
  - 1 次グループ ID の変更
    - newgrp コマンドを使用する 48
- ゲーム
  - クイズ 639
  - 数値書き込みゲーム 343
- 経路指定テーブル、ネットワーク
  - 管理
    - routed デーモンを使用する 872
- 構成
  - NFS ネットワーク変数
    - nfso コマンドを使用する 62
  - NIM クライアント・パッケージ
    - niminit コマンドを使用する 150
- 構成データベース
  - ローカル resolver ルーチン用にドメイン・ネーム・サーバー・エントリーを操作する
    - namerslv コマンドを使用する 7
- コマンド
  - イベント応答リソース・マネージャー (ERRM)
    - notifyevent 313



## コマンド (続き)

の異なる優先順位の設定

nice コマンドを使用する 89

ハングアップせずに実行 311

namerslv 7

netstat 33

newform 45

nfso 62

nim 91

nimconfig 142

niminit 150

niscat 181

nisdefaults 192

nistbladm 223

nlsrc 235

nmon 241

no 260

notifyevent 313

ntpq 336

oslevel 363

passwd 377

pic 413

pkgtrans 472

pr 504

preparevsd 509

preprnode 511

printf 515

prs 552

ps 561

psroff 591

qdaemon 616

quot 641

quota 643

quotacheck 644

quotaoff 646

quotaon 646

ras\_logger 660

rc 662

rcp 666

rc.powerfail 664

rdist 675

rdump 689

read 692

refsrc 708

refsensor 710

removevsd 717

remsh 892

resumevsd 749

rev 750

rmaudrec 774

rmcctrl 780

rmcomg 787

## コマンド (続き)

rmcondition 789

rmcondresp 791

rmddel 796

rmdev 797

rmdir 799

rmlpcmd 809

rmm 815

rmnamsv 817

rmnfsexp 818

rmramdisk 830

rmresponse 831

rmrpdomain 835

rmrpnode 837

rmrsrc 841

rmsensor 845

rmss 849

rmvirprt 861

Rsh 890

rsh 892

runact 905

runlpcmd 910

ruser 916

rvsdrestrict 920

## [サ行]

### サーバー機能

リモート・コマンドの実行、提供する

rshd デーモンを使用する 896

### サービス要求、処理

PC-NFS クライアントから

pcnfsd デーモンを使用する 880

### 索引

主題ページの索引の作成

ndx コマンドを使用する 19

順列索引の生成

ptx コマンドを使用する 596

### サブサーバー

SRC オブジェクト定義の除去 847

### サブシステム

通知メソッドの除去 821

のリフレッシュの要求

refresh コマンドを使用する 707

SRC オブジェクト・クラスからの定義の除去 853

### 参考文献データベース

印刷

roffbib コマンドを使用する 866

### シェル

制限付きの 890

Bourne 890

- シェル変数
  - 標準入力フィールドとの比較
    - read コマンドを使用する 692
- 磁気テープ・デバイス
  - リモート・アクセスの可能化
    - rmt コマンド 855
- システム
  - からのデバイスの除去
    - rmdev コマンドを使用する 797
  - リブート
    - reboot コマンドを使用する 695
- システム統計
  - 記録
    - nmon コマンド 241
- システムのブート
  - ブート・イメージ
    - からの情報の読み取り 732
- システムをシミュレートする
  - パフォーマンス・テスト
    - rmss コマンドを使用する 849
- システム/370 ホスト・インターフェース・アダプター
  - 正しくインストールされていることの検査 376
- システム・シャットダウン
  - 電源障害の検出時にシステムをシャットダウン
    - rc.powerfail コマンドを使用する 664
- システム・テーブル
  - の内容の解釈 595
- システム・リソース・コントローラー 821, 853
- 始動
  - 標準の始動初期化の実行
    - rc コマンドを使用する 662
- 出力、標準
  - 書き込み
    - pr コマンドを使用する 504
- 順列索引
  - 生成
    - ptx コマンドを使用する 596
- 照会プログラム
  - Network Time Protocol を開始する
    - ntpq コマンドを使用する 336
- 状況
  - プロセス、表示
    - ps コマンドを使用する 561
- 初期化
  - NIM マスター・パッケージ
    - nimconfig コマンドを使用する 142
- 初期化、始動
  - rc コマンドを使用する 662
- 除去
  - ディレクトリー
    - rmdir コマンドを使用する 799

- 所有権
  - ファイルシステムの要約
    - quot コマンドを使用する 641
- 数式テキスト
  - フォーマット
    - neqn コマンドを使用する 21
- スクリプト
  - イベント応答リソース・マネージャー (ERRM)
    - notifyevent 313
  - notifyevent 313
- 制限付きのシェル
  - 起動
    - Rsh コマンドを使用する 890
- ソース・ファイル 383

## [夕行]

- 端末
  - 初期化
    - reset コマンドを使用する 728
  - 特性の設定
    - reset コマンドを使用する 728
- 端末の設定
  - 現行ウィンドウ・サイズへの設定
    - resize コマンドを使用する 730
- 遅延ログイン・ポート 405
- 着信メッセージ
  - 追加の宛先へのコピーの送信 671
  - ユーザーへの通知 674
- チューニング
  - ネットワーク・パラメーター
    - no コマンドを使用する 260
- 通知オブジェクト・クラス (SRC)
  - 通知メソッド定義の除去 821
- データベース、システム
  - 外部ホスト・アクセスの制御、操作
    - ruser コマンドを使用する 916
- テーブル、ネットワーク経路指定
  - 管理
    - routed デーモンを使用する 872
- テーブル、NIS+
  - 表示
    - niscat コマンドを使用する 181
- デーモン
  - ndpd-router 14
  - nfsrgyd 84
  - nrglbd 314
  - penfsd 880
  - pppattachd 491
  - routed 872
  - rshd 896

- ディスク使用率
  - クォータを表示する
    - quota コマンドを使用する 643
- ディスプレイ
  - ファイル
    - 画面に合わせたフォーマット 409
- ディレクトリー
  - 削除 766
  - 除去 766
    - rmdir コマンドを使用する 799
  - のパス名
    - 表示 601
- ディレクトリー、アンエクスポート
  - NFS クライアントから
    - rmnfsexp コマンドを使用する 818
- テキスト
  - のフォーマットの変更
    - newform コマンドを使用する 45
- テスト、パフォーマンス
  - システムをシミュレートする
    - rmss コマンドを使用する 849
- デバイス
  - システムからの除去
    - rmdev コマンドを使用する 797
- デバイス構成コマンド
  - restbase 732
- デフォルト値
  - ネーム・スペース内でアクティブな、7 つを表示する
    - nisdefaults コマンドを使用する 192
- デルタ・ファイル
  - 除去
    - rmddel コマンドを使用する 796
- 電源障害
  - 起こった場合にシステムをシャットダウンする
    - rc.powerfail コマンドを使用する 664
- 動的ホスト構成プロトコル
  - DNS サーバーを更新する
    - nsupdate コマンド 324
    - nsupdate4 コマンド 325
- ドメイン・ネーム・サーバー
  - 構成 DB 内のローカル resolver ルーチンのエントリ一、操作
    - namerslv コマンドを使用する 7
- ドライバ
  - プリンターのフォーマット
    - pioformat コマンドを使用する 439

## [ナ行]

- ネーム・サービス
  - TCP/IP を構成解除する
    - rmnamsv コマンドを使用する 817

- ネーム・スペース
  - アクティブな 7 つのデフォルト値を表示する
    - nisdefaults コマンドを使用する 192
- ネットワーク経路指定テーブル
  - 管理
    - routed デーモンを使用する 872
- ネットワーク状況
  - 表示
    - netstat コマンドを使用する 33
- ネットワークの CPU 使用量 22
- ネットワーク変数
  - NFS を構成する
    - nfso コマンドを使用する 62
- ネットワーク・インストール管理
  - 操作
    - nim コマンドを使用する 91
- ネットワーク・インストール・マネージャー 139, 142
  - クライアント・パッケージを構成する
    - niminit コマンドを使用する 150
  - マスター・パッケージを初期化する
    - nimconfig コマンドを使用する 142
- ネットワーク・パラメーター
  - チューニング
    - no コマンドを使用する 260
- ネットワーク・ファイルシステム
  - ネットワーク変数を構成する
    - nfso コマンドを使用する 62

## [ハ行]

- パーソナル・プリンター・データ・ストリーム
  - を扱うプリンター用のテキストのフォーマット
    - proff コマンドを使用する 537
- パス名
  - 表示 601
- パスワード
  - に関するユーザー・データベース内の情報の検証
    - pwdck コマンドを使用する 604
  - 変更
    - passwd コマンドを使用する 377
  - ユーザー・パスワードの管理
    - pwdadm コマンドを使用する 602
- パッケージ
  - インストール可能物の作成
    - pkgmk コマンドを使用する 465
- パッケージ・フォーマット
  - 変換
    - pkgtrans コマンドを使用する 472
- パフォーマンス・テスト
  - システムをシミュレートする
    - rmss コマンドを使用する 849

- ピクチャー、描画
  - troff コマンド入力のプリプロセス
  - pic コマンドを使用する 413
- ピクチャーの描画
  - troff コマンド入力のプリプロセス
  - pic コマンドを使用する 413
- 非同期接続
  - PPP サブシステム
    - pppdial コマンド 499
- 非同期デバイス・ストリーム
  - PPP サブシステムへの接続
    - pppattachd デーモンを使用する 491
- 表示
  - ネットワーク状況
    - netstat コマンドを使用する 33
- 標準出力
  - へのフォーマット済み出力の書き出し
    - printf コマンドを使用する 515
- 標準入力
  - 1 行を読み取る
    - read コマンドを使用する 692
- ブート・イメージ 732
- ファイル
  - 圧縮
    - pack コマンドを使用する 369
  - アンパック
    - pcat コマンドを使用する 404
  - 行の番号付け 233
  - 削除 766
  - 指定されたフォーマットでの表示 344
  - 除去 766
  - ディスプレイに合わせたフォーマット 409
  - 内容の表示 409
  - の各行の文字の順序を逆にする
    - rev コマンドを使用する 750
  - パス名の検査
    - pathchk コマンドを使用する 388
  - 標準出力への書き出し
    - pr コマンドを使用する 504
  - フォルダーの内容を圧縮して 1 つのファイルに入れる
    - packf コマンドを使用する 371
  - 複数の行のマージ
    - paste コマンドを使用する 380
  - リモート・マシンのデバイス上へのバックアップ
    - rdump コマンドを使用する 689
  - ローカル・ホストとリモート・ホスト間の転送
    - rcp コマンドを使用する 666
  - 1 つのファイルの行のマージ
    - paste コマンドを使用する 380
  - 2 つのリモート・ホスト間の転送
    - rcp コマンドを使用する 666

- ファイル (続き)
  - backup コマンドで作成されたファイルのコピー
    - restore コマンドを使用する 733
  - i ノード番号からのパス名の生成
    - ncheck コマンドを使用する 10
  - SCCS を表示する
    - prs コマンドを使用する 552
- ファイルシステム
  - クォータ、オン/オフの切り替え
    - quotaon または quotaoff コマンドを使用する 646
  - クォータの整合性、検査
    - quotacheck コマンドを使用する 644
  - 所有権、要約
    - quot コマンドを使用する 641
  - プロトタイプ・ファイルの構成
    - proto コマンドを使用する 549
  - リモート・マシンからローカル・マシンへのバックアップのコピー
    - rrestore コマンドを使用する 887
- ファイルのコピー、同一の
  - 複数ホスト、配布
    - rdist コマンドを使用する 675
- ファイルの同一コピー
  - 複数ホスト、配布
    - rdist コマンドを使用する 675
- ファイルの同一コピーを配布する
  - 複数ホスト
    - rdist コマンドを使用する 675
- ファイルをバックアップする
  - リモート・マシンのデバイス上へ
    - rdump コマンドを使用する 689
- ファイルを表示する
  - SCCS
    - prs コマンドを使用する 552
- フィールド、標準入力
  - シェル変数との比較
    - read コマンドを使用する 692
- フォーマット
  - テキストを変更する
    - newform コマンドを使用する 45
- フォーマット済み出力
  - 標準出力への書き出し
    - printf コマンドを使用する 515
- フォルダー
  - 圧縮して 1 つのファイルに入れる
    - packf コマンドを使用する 371
  - 削除
    - rmf コマンドを使用する 801
  - に含まれるメッセージの除去
    - rmf コマンドを使用する 801

- フォント
  - コピー
    - piofontin コマンドを使用する 438
- 複数ホスト
  - ファイルの同一コピーを配布する
    - rdist コマンドを使用する 675
- 物理ボリューム 725
- プリンター
  - 上のジョブの開始
    - qprt コマンドを使用する 628
  - 上のジョブのスケジュール
    - qdaemon コマンドを使用する 616
  - 上のジョブの取り消し
    - qcan コマンドを使用する 613
  - 事前定義データ・ストリーム定義の作成
    - piopredef コマンドを使用する 454
  - 新規の追加 448
  - スプーリング・システムの状況の提供
    - qstatus コマンドを使用する 637
  - バースト・ページ
    - pioburst コマンド 432
  - フォント
    - コピー 438
- プリンター、仮想
  - 除去
    - rmvirprt コマンドを使用する 861
- プリンター定義ファイル
  - 拡張または縮小 434
- プリンターの定義
  - 仮想プリンターの 435
- プリンターのバックエンド・コマンド
  - piodmng 437
  - piolsvp 442
  - piomgpdev 445
  - piomkapqd 446
  - piomkpbq 448
  - piomsg 450
- プリンター・キュー
  - システムからの除去
    - rmque コマンドを使用する 828
    - rmquedev コマンドを使用する 829
  - の状況の表示
    - qchk コマンドを使用する 614
- プリンター・ジョブ
  - enq コマンドによりエンキューされた、スケジュールング
    - qdaemon コマンドを使用する 616
- プリンター・スプーリング・システム
  - のシステム管理機能の実行
    - qadm コマンドを使用する 611
- プリンター・バックエンド
  - へのメッセージの送信 450
- プリンター・フォーマッター
  - 始動
    - pioformat コマンドを使用する 439
- プロセス
  - 実行中のプロセスの優先順位の変更
    - renice コマンドを使用する 718
  - の状況の表示
    - ps コマンドを使用する 561
- プロッター・キュー
  - システムからの除去
    - rmquedev コマンドを使用する 829
- プロット、HP-GL ファイルの 476, 478
- プロット・ファイル
  - PostScript への変換
    - psplot コマンドを使用する 589
- プロンプト・エディターの始動、MH 547
- 文書
  - 内の参考資料の検出
    - refer コマンドを使用する 702
  - 内への参考資料の挿入
    - refer コマンドを使用する 702
- ページング・スペース
  - 除去 825
- 別名データベース
  - 作成 45
- 変換
  - パッケージ・フォーマット
    - pkgtrans コマンドを使用する 472
- ポート
  - ログイン 406
- 保守およびテクノロジー・レベル、システム
  - レポート、最後にインストールされた
    - oslevel コマンドを使用する 363
- ホスト
  - の状況の表示
    - ruptime コマンドを使用する 915
  - リモート・ホスト上でのコマンドの実行
    - rsh コマンドを使用する 892
  - リモート・ホストへのログイン
    - rsh コマンドを使用する 892
  - ローカルのリモートとの接続
    - rlogin コマンドを使用する 761
- ホスト、複数
  - ファイルの同一コピーを配布する
    - rdist コマンドを使用する 675
- ボリューム・グループ
  - 物理区画割り当ての再編成
    - reorgvg コマンドを使用する 720
  - 物理ボリュームの除去
    - reducevg コマンドを使用する 700
  - ボリューム・グループの再作成 696

ボリューム・グループ (続き)  
論理ボリュームの除去  
rmlv コマンドを使用する 812

## [マ行]

メールのフォルダーへの取り込み、MH 672  
メール・コマンド  
newaliases 45  
rmail 773  
メッセージ  
アクティブ状況からの除去  
rmm コマンドを使用する 815  
他のフォルダーへのファイリング  
refile コマンドを使用する 704  
直前のメッセージの表示  
prev コマンドを使用する 513  
次のメッセージの表示  
next コマンドを使用する 52  
内容による選択  
pick コマンドを使用する 420  
に対する応答  
repl コマンドを使用する 722  
バック・ファイルへの保存 672  
標準出力への書き出し 51  
メッセージ機能コマンド  
runcat 909  
メッセージ経路指定 489  
メッセージを除去する  
アクティブ状況から  
rmm コマンドを使用する 815  
メッセージ・シーケンス  
作成  
pick コマンドを使用する 420  
変更  
pick コマンドを使用する 420  
メッセージ・ソース・ファイル  
作成 909  
メッセージ・ソース・ファイルの作成 909  
メモリー、各種サイズの  
アプリケーションのパフォーマンスをテストする、シ  
ステムをシミュレートする  
rmss コマンドを使用する 849  
メモリー管理  
システムのページ・サイズの表示 374  
文字  
ファイルの各行の文字の順序を逆にする  
rev コマンドを使用する 750

## [ヤ行]

ユーザー  
アカウントの除去  
rmuser コマンドを使用する 859  
に関するパスワード情報の検証  
pwdck コマンドを使用する 604  
認証 375  
のパスワードの管理  
pwdadm コマンドを使用する 602  
パスワードの変更  
passwd コマンドを使用する 377  
ユーザーの認証 375  
読み取り  
1 行の標準入力  
read コマンドを使用する 692

## [ラ行]

ライブラリー  
アーカイブ・ライブラリーの変換 651  
ライン・プリンター  
用のテキストのフォーマット  
nroff コマンドを使用する 315  
リモート・コマンドの実行  
サーバー機能を提供する  
rshd デーモンを使用する 896  
リモート・マシンのデバイス  
ファイルをバックアップする  
rdump コマンドを使用する 689  
リンク  
除去 766  
ルーター  
NDP および RIPng デーモン  
ndpd-router デーモンを使用する 14  
レポート  
最後にインストールされた保守およびテクノロジー・  
レベル  
oslevel コマンドを使用する 363  
ローカル resolver ルーチン  
構成 DB 内の、ドメイン・ネーム・サーバー・エン  
トリーの操作  
namerslv コマンドを使用する 7  
ローカル・システム統計  
対話モードでの表示  
nmon コマンド 241  
ログイン・ポート  
共用  
使用可能化 588  
リスト 588  
使用可能化  
pdelay コマンドを使用する 405

ログイン・ポート (続き)

使用可能化 (続き)

penable コマンドを使用する 407

pshare コマンドを使用する 588

pstart コマンドを使用する 594

使用不可

pdisable コマンドを使用する 406

phold コマンドを使用する 412

遅延

使用可能化 405

リスト 405

リスト

pdelay コマンドを使用する 405

penable コマンドを使用する 407

phold コマンドを使用する 412

pstart コマンドを使用する 594

論理ボリューム

あるボリュームの新規ボリュームへのコピー 699  
からのミラーの除去

rmlvcopy コマンドを使用する 814

ボリューム・グループからの除去

rmlv コマンドを使用する 812

## A

acct/\* コマンド

削除 716

nulladm 343

prctmp 507

prdaily 508

prtacct 556

## B

Bourne シェル shell

制限付きバージョンを呼び出す

Rsh コマンドを使用する 890

## C

C プログラミング言語

パターンの宣言へのコンパイル 713

## D

Diablo 630 印刷ファイル

PostScript への変換

ps630 コマンドを使用する 584

diff リスト 383

## E

enq コマンド

エンキューされたジョブのスケジュール

qdaemon コマンドを使用する 616

ERRM コマンド

notifievent 313

ERRM スクリプト

elogevent 313

## F

fastboot コマンド 695

## H

HCON

システム/370 ホスト・インターフェース・アダプタ  
ー

アクティビティの診断 376

## I

IMAP コマンド

pop3d 483

pop3ds 485

IPv6 NDP (隣接ディスカバリー・プロトコル) 12

## K

Korn シェル

拡張 758

起動 758

制限付きの 758

## M

MH

post コマンド 489

prompter コマンド 547

rcvdist コマンド 671

rcvpack コマンド 672

rcvstore コマンド 672

rcvtty 674

MultiPath I/O

rmpath コマンド 822

## N

named デーモン 1

の説明 1

named9 5



- namerslv コマンド 7
- ncheck コマンド 10
- NCS デーモン
  - nrglbd 314
- nddctl コマンド 11
- NDP および RIPng デーモン
  - ルーター用の
    - ndpd-router デーモンを使用する 14
- ndp コマンド 12
- ndp デーモン 12
- NDP (隣接ディスカバリー・プロトコル) 12
- ndpd-router デーモン 14
- ndx コマンド 19
- neqn コマンド 21
- nesgrep 情報
  - 検索 195
- netpmon コマンド 22
- netstat コマンド 33
  - インターフェース表示 37
  - 経路指定テーブル表示 37
- Network Time Protocol
  - 開始する
    - ntpq コマンドを使用する 336
- Network Time Protocol コマンド
  - ntpdate 334
  - ntptrace 341
- newaliases コマンド
  - メール 45
- newform コマンド 45
- newgrp コマンド 48
- newkey コマンド
  - NIS 49
- next コマンド 52
- NFS
  - ネットワーク変数を構成する
    - nfso コマンドを使用する 62
- NFS クライアント
  - ディレクトリーのアンエクスポート
    - rmnfsexp コマンドを使用する 818
- NFS コマンド
  - nfsstat 85
  - on 357
  - rmnfs 817
  - rmnfsexp 818
  - rmnfsmnt 819
  - rpcgen 882
  - rpcinfo 884
  - rup 914
  - rusers 918
  - rwall 922
- NFS デーモン
  - nfsd 58
- NFS デーモン (続き)
  - pcnfsd 880
  - portmap 486
  - rexid 752
  - rstatd 899
  - rusersd 920
  - rwalld 923
- nfs4cl コマンド 55
- nfs4smctl 57
- nfsd デーモン 58
- nfshostkey 60
- nfshostmap 61
- nfso コマンド 62
- nfsrgyd 84
- nfsstat コマンド 85
- nfs.clean コマンド 54
- nice コマンド 89
- NIM
  - クライアント・パッケージを構成する
    - niminit コマンドを使用する 150
    - マスター・パッケージを初期化する
      - nimconfig コマンドを使用する 142
- NIM オブジェクト
  - 操作を実行する
    - nim コマンドを使用する 91
- NIM コマンド
  - nim 91
  - nimadapters 124
  - nimclient 139
  - nimconfig 142
  - niminit 150
  - nim\_clients\_setup 104
  - nim\_master\_recover 106
  - nim\_master\_setup 109
  - nim\_update\_all 122
- nim コマンド 91
- nimadapters 124
- nimadm コマンド 130
- nimclient コマンド 139
- nimconfig コマンド 142
- nimdef コマンド 145
- niminit コマンド 150
- niminv コマンド 153
- nimol\_backup コマンド 160
- nimol\_config コマンド 162
- nimol\_install コマンド 165
- nimol\_lslpp コマンド 167
- nimol\_update コマンド 169
- nimquery 170
- nim\_clients\_setup 104
- nim\_master\_recover 106
- nim\_master\_setup 109

nim\_move\_up コマンド 112  
nim\_update\_all 122  
NIS  
    コマンド  
        nistest 227  
NIS コマンド  
    newkey 49  
    rmkeyserv 808  
    rmyp 863  
    rm\_niscachemgr 769  
    rm\_nisd 771  
    rm\_nispasswd 772  
niscat コマンド 181  
nisdefaults コマンド 192  
nisgrep コマンド 195  
nistbladm コマンド 223  
nistest コマンド  
    の説明 227  
nistoldif 229  
NIS+  
    テーブル、表示  
        niscat コマンドを使用する 181  
NIS+ テーブル  
    管理  
        nistbladm コマンドを使用する 223  
    表示  
        niscat コマンドを使用する 181  
nlssrc コマンド 235  
nmon コマンド 241  
no コマンド 260  
nohup コマンド 311  
notifyevent コマンド 313  
notifyevent スクリプト 313  
nrglbd  
    の説明 314  
nroff コマンド 315  
    の数式テキストのフォーマット  
        neqn コマンドを使用する 21  
nslookup コマンド 318  
nsupdate コマンド 324  
nsupdate4 コマンド 325  
nsupdate9 330  
NTP  
    開始する  
        ntpq コマンドを使用する 336  
ntpdate コマンド 334  
ntpq コマンド 336  
ntptrace コマンド 341  
nulladm コマンド 343

## O

ODM (オブジェクト・データ・マネージャー)  
    オブジェクト  
        オブジェクト・クラスからの基準にマッチするオブジェクトの検索 356  
        オブジェクト・クラスからの除去 354  
        オブジェクト・クラス内の変更 350  
        オブジェクト・クラスへの追加 349  
    オブジェクト・クラス  
        オブジェクトの除去 354  
        記述の表示 357  
        基準にマッチするオブジェクトの検索 356  
        コンパイル 351  
        削除 355  
        内のオブジェクトの変更 350  
        へのオブジェクトの追加 349  
        データベースを圧縮する 437  
on コマンド 357  
oslevel コマンド 363  
OS\_install コマンド 359

## P

pac コマンド 367  
pack コマンド 369  
packf コマンド 371  
pagdel 373  
paginit 375  
paglist 376  
panel20 コマンド 376  
passwd コマンド 377  
paste コマンド 380  
patch コマンド 383  
pathchk コマンド 388  
pax コマンド  
    アーカイブの取り出し、書き込み、およびリストとファイルのコピー 389  
pcat コマンド 404  
pcnfsd デーモン 880  
PC-NFS クライアント  
    サービス要求の処理  
        pcnfsd デーモンを使用する 880  
pdelay コマンド 405  
pdisable コマンド 406  
penable コマンド 407  
perfwb コマンド 408  
phold コマンド 412  
pic コマンド 413  
pick コマンド 420  
ping コマンド  
    の説明 424

ping コマンド (続き)  
 の例 426

piobe コマンド 430

pioburst コマンド 432

piocnvt コマンド 428, 434

piodigest コマンド 435

piofontin コマンド 438

pioformat コマンド 439

piofquote コマンド 441

pioout コマンド 452

piopredef コマンド 454

pkgmk コマンド 465

pkgtrans コマンド 472

platform\_dump 474

plotgbe コマンド 476

plotlbe コマンド 478

pmlist コマンド 480

pmtu コマンド 482

pop3d デーモン 483

pop3ds デーモン 485

portmap デーモン 486

portmir コマンド 487

post コマンド 489

PostScript ファイル  
 印刷のためにページ順序を逆にする  
 psrev コマンドを使用する 590  
 印刷のページ範囲を選択する  
 psrev コマンドを使用する 590  
 Diablo 630 ファイルからの変換  
 ps630 コマンドを使用する 584  
 Tektronix 4014 ファイルからの変換  
 ps4014 コマンドを使用する 583  
 troff からの変換  
 psroff コマンドを使用する 591  
 troff 中間ファイルからの変換  
 psc コマンドを使用する 585  
 psdit コマンドを使用する 585

PostScript プリンター  
 用の特定の制御文字の変換  
 piofquote コマンドを使用する 441

PPP 493

PPP サブシステム  
 始動と管理  
 pppcontrold デーモン 493  
 接続された非同期デバイス・ストリーム  
 pppattachd デーモンを使用する 491  
 非同期接続 499  
 RAS 情報を表示する  
 pppstat コマンド 501

pppattachd デーモン 491

pppcontrold デーモン 493

pppdial コマンド 499

pppstat コマンド 501

pr コマンド 504

prctmp コマンド 507

prdaily コマンド 508

preparevsd コマンド 509

preprnode コマンド 511

prev コマンド 513

printf コマンド 515

proccred コマンド 519

procfiles コマンド 520

procflags コマンド 522

procldd コマンド 523

procmap コマンド 524

procrun コマンド 526

procsig コマンド 527

procstack コマンド 528

procstop コマンド 530

proctree コマンド 531

procwait コマンド 532

procwdx コマンド 533

proff コマンド 537

projctl コマンド 538

prompter コマンド 547

proto コマンド 549

prs コマンド 552

prtacct コマンド 556

prtconf 557

ps コマンド 561

ps4014 コマンド 583

ps630 コマンド 584

psc コマンド 585

psdit コマンド 585

pshare コマンド 588

psplot コマンド 589

psrev コマンド 590

psroff コマンド 591

pstart コマンド 594

pstat コマンド 595

ptx コマンド 596

pwchange コマンド 598

pwck コマンド 600

pwdadm コマンド 602

pwdck コマンド 604

pwtokey コマンド 607

## Q

qadm コマンド 611

qcan コマンド 613

qchk コマンド 614

qdaemon コマンド 616

qhld コマンド 618

qmov コマンド 619  
 qosstat 625  
 qpri コマンド 627  
 qpri コマンド 628  
 qstatus コマンド 637  
 quot コマンド 641  
 quota  
   整合性、ファイルシステムの検査  
   quotacheck コマンドを使用する 644  
 quota コマンド 643  
 quotacheck コマンド 644  
 quotaoff コマンド 646  
 quotaon コマンド 646  
 quotas  
   ディスク使用率を表示する  
   quota コマンドを使用する 643  
   ファイルシステムのオン/オフ  
   quotaon または quotaoff コマンドを使用する 646

## R

raddbm コマンド 647  
 RAM ディスク 830  
 RAM ディスク、仮想  
   除去  
   rmramdisk コマンドを使用する 830  
 raso コマンド 652  
 ras\_logger コマンド 660  
 rc コマンド 662  
 rcp コマンド 666  
 rcvdist コマンド 671  
 rcvpack コマンド 672  
 rcvstore コマンド 672  
 rcvty コマンド 674  
 rc.mobip6 663  
 rc.powerfail コマンド 664  
 rdist コマンド 675  
 rdistd コマンド 689  
 rdump コマンド 689  
 read コマンド 692  
 reboot コマンド 695  
 recreatevg コマンド 696  
 recsh コマンド 698  
 redefinevg コマンド 699  
 reducevg コマンド 700  
 refer コマンド 702  
 refile コマンド 704  
 refresh コマンド 707  
 reffsrc コマンド 708  
 refsensor コマンド 710  
 remove コマンド 716  
 removevsd コマンド 717  
 remsh コマンド 892  
 renice コマンド 718  
 reorgvg コマンド 720  
 repl コマンド 722  
 replacepv コマンド 725  
 repquota コマンド 727  
 reset コマンド 728  
 resize コマンド 730  
 restart-secldapclntd 731  
 restbase コマンド 732  
 restore コマンド 733  
 restvg コマンド 746  
 resumevsd コマンド 749  
 rev コマンド 750  
 rexd デーモン 752  
 rexec コマンド 753  
 rexec コマンド用のサーバー機能、TCP/IP 754  
 rexecd デーモン 754  
 rgb コマンド 756  
 RIPng および NDP デーモン  
   ルーター用の  
   ndpd-router デーモンを使用する 14  
 rksh コマンド 758  
 rlogin コマンド 761  
 rlogind デーモン 764  
 rmail コマンド 773  
 rmaudrec コマンド 774  
 rmC2admin コマンド 778  
 rmCCadmin コマンド 779  
 rmccctrl コマンド 780  
 rmcifscrd コマンド 783  
 rmcifsmnt コマンド 784  
 rmclass コマンド 785  
 rmcomg コマンド 787  
 rmcondition コマンド 789  
 rmcondresp コマンド 791  
 rmcosi コマンド 795  
 rmdel コマンド 796  
 rmdev コマンド 797  
 rmdir コマンド 799  
 rmf コマンド 801  
 rmfs コマンド 803  
 rmgroup コマンド 804  
 rmiscsi コマンド 806  
 rmitab コマンド 808  
 rmkeyserv コマンド 808  
 rmlpcmd コマンド 809  
 rmlv コマンド 812  
 rmlvcopy コマンド 814  
 rmm コマンド 815  
 rmnamsv コマンド 817  
 rnmfs コマンド 817

rmnfsexp コマンド 818  
rmnfsmnt コマンド 819  
rmnfsproxy コマンド 820  
rmpath コマンド 822  
rmprtsv コマンド 824  
rmpps コマンド 825  
rmque コマンド 828  
rmquedev コマンド 829  
rmramdisk コマンド 830  
rmresponse コマンド 831  
rmrole コマンド 834  
rmrpdomain コマンド 835  
rmrpnode コマンド 837  
rmrset 840  
rmrsrc コマンド 841  
rmsensor コマンド 845  
rmss コマンド 849  
rmts コマンド 857  
rmuser コマンド 859  
rmvfs コマンド 861  
rmvirprt コマンド 861  
rmyp コマンド 863  
rm\_niscachemgr コマンド 769  
rm\_nisd コマンド 771  
rm\_nispasswd コマンド 772  
rndc 864  
rndc-confgen コマンド 865  
roffbib コマンド 866  
rollback コマンド 867  
route コマンド 868  
routed デーモン 872  
    経路指定テーブルの更新 873  
    シグナル 875  
    gateways ファイルのフォーマット 874  
    SRC による操作 875  
rpcgen コマンド 882  
rpcinfo コマンド 884  
rrestore コマンド 887  
Rsh コマンド 890  
rsh コマンド 892  
rshd デーモン 896  
rstatd デーモン 899  
rt\_enable コマンド 900  
runacct コマンド 902  
runact コマンド 905  
runcat コマンド 909  
runlpcmd コマンド 910  
rup コマンド 914  
ruptime コマンド 915  
ruser コマンド 916  
rusers コマンド 918  
rusersd デーモン 920

rvsdrestrict コマンド 920  
rwall コマンド 922  
rwalld デーモン 923  
rwho コマンド 924  
rwhod コマンド  
    SRC による操作 926  
rwhod デーモン 925

## S

### SCCS

    デルタ・ファイルを除去する  
        rmdel コマンドを使用する 796  
    ファイル、表示  
        prs コマンドを使用する 552

### SCCS コマンド

    prs 552  
    rmdel 796

### SMIT

    による印刷キューの作成 446  
    によるプリンターの作成 446  
    プリンター・ダイアログの構築 446

### SRC

    サブサーバー・オブジェクト定義の除去 847  
    サブシステム通知メソッドの除去 821  
    サブシステム・オブジェクト定義の除去 853

### SRC 構成コマンド

    rmnotify 821  
    rmserver 847  
    rmssys 853

## T

### TCP/IP

    印刷サービス  
        構成解除 824  
    インターネット・ドメイン・ネーム・サーバーの照会  
        318  
    経路指定テーブル  
        手動入力の実行 868  
    経路指定テーブルの管理  
        routed デーモンを使用する 872  
    構成データベース  
        ネーム・サーバーのエントリーの制御 7  
    構成データベース管理エントリー  
        ruser コマンドを使用する 916  
    コマンド  
        リモート・ホスト上での実行 753  
    サーバー機能  
        提供 764  
        のサポート 925

## TCP/IP (続き)

サーバー機能を提供する

rshd デーモンを使用する 896

デーモン

named 1

rexecd 754

ネーム・サービスを構成解除する

rmnamsv コマンドを使用する 817

ネットワーク状況を入力する

netstat コマンドを使用する 33

パラメーター

チューニング 260

ホスト

ログインしているユーザーのリスト表示 924

## TCP/IP smit コマンド

namerslv 7

rmnamsv 817

rmprtsv 824

ruser 916

## TCP/IP コマンド

namerslv 7

netstat 33

no 260

nslookup 318

rmnamsv 817

rmprtsv 824

route 868

ruser 916

rwho 924

## TCP/IP デーモン

rlogind 764

routed 872

rshd 896

rwhod 925

## Tektronix 4014 ファイル

PostScript への変換

ps4014 コマンドを使用する 583

## termcap 環境変数

現行ウィンドウ・サイズへの設定

resize コマンドを使用する 730

## troff コマンド

組版システムで図を描くためのプリプロセッサ

pic コマンドを使用する 413

## troff 中間ファイル・フォーマット

PostScript フォーマットへの変換

psc コマンドを使用する 585

psdit コマンドを使用する 585

## troff ファイル

PostScript への変換

psroff コマンドを使用する 591

## [特殊文字]

/etc/filesystems ファイル

エントリーの除去

rmfs コマンドを使用する 803

/etc/inittab ファイル

レコードの除去

rmitab コマンドを使用する 808

/etc/vfs ファイル

エントリーの除去

rmvfs コマンドを使用する 861







Printed in Japan

SC88-6892-07



日本アイ・ビー・エム株式会社  
〒103-8510 東京都中央区日本橋箱崎町19-21