

AIX バージョン 7.2

**コマンド・リファレンス
第 4 巻 (n から r)**

IBM

AIX バージョン 7.2

**コマンド・リファレンス
第 4 巻 (n から r)**

IBM

注記

本書および本書で紹介する製品をご使用になる前に、943 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 バージョン 7.2 および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

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： AIX Version 7.2

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

© Copyright IBM Corporation 2015, 2017.

目次

本書について	vii
強調表示	vii
AIX での大/小文字の区別	vii
ISO 9000	vii
Single UNIX Specification のサポート	viii
n	1
named デーモン	1
named-checkconf コマンド	2
named-checkzone、named-compilezone コマンド	2
named8 デーモン	4
named9 デーモン	8
namerslv コマンド	9
ncheck コマンド	12
nddctl コマンド	13
ndp コマンド	14
ndpd-host デーモン	15
ndpd-router デーモン	17
ndx コマンド	22
neqn コマンド	24
netcd デーモン	25
netcdctl コマンド	27
netpmon コマンド	29
netrule コマンド	39
netstat コマンド	44
newaliases コマンド	55
newform コマンド	56
newgrp コマンド	58
newkey コマンド	60
news コマンド	61
next コマンド	63
nfs.clean コマンド	64
nfs4cl コマンド	65
nfs4smctl コマンド	67
nfsauthreset コマンド	68
nfsd デーモン	69
nfsdhostkey コマンド	71
nfsdhostmap コマンド	72
nfso コマンド	72
nfsrgyd デーモン	81
nfsstat コマンド	82
nice コマンド	86
nim コマンド	88
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	123
nimadapters コマンド	125
nimadm コマンド	131
nimclient コマンド	140

nimconfig コマンド	144
nimdef コマンド	147
niminit コマンド	151
niminv コマンド	155
nimol_backup コマンド	161
nimol_config コマンド	163
nimol_install コマンド	165
nimol_lslpp コマンド	168
nimol_update コマンド	169
nimquery コマンド	171
nistoldif コマンド	172
nl コマンド	175
nlsrc コマンド	177
nm コマンド	179
nmon コマンド	182
no コマンド	203
nohup コマンド	227
enotifyevent コマンド、notifyevent コマンド	229
nroff コマンド	230
nslookup コマンド	233
nsupdate コマンド	236
nsupdate4 コマンド	237
nsupdate8 コマンド	239
nsupdate9 コマンド	241
ntpd4 デーモン	245
ntpdate コマンド	248
ntpdate4 コマンド	250
ntpdc4 コマンド	253
ntp-keygen4 コマンド	259
ntpq コマンド	263
ntpq4 デーモン	268
ntptrace コマンド	272
ntptrace4 コマンド	274
nulladm コマンド	276
number コマンド	276
o	279
od コマンド	279
odmadd コマンド	283
odmchange コマンド	285
odmcreate コマンド	285
odmdelete コマンド	287
odmdrop コマンド	288
odmget コマンド	289
odmshow コマンド	289
on コマンド	290
openpts コマンド	291
OS_install コマンド	292
oslevel コマンド	298
ospf_monitor コマンド	300

p	303
pac コマンド	303
pack コマンド	304
packf コマンド	306
pagdel コマンド	308
pagesize コマンド	309
paginit コマンド	310
paglist コマンド	311
panel20 コマンド	312
passwd コマンド	312
paste コマンド	316
patch コマンド	318
pathchk コマンド	323
pax コマンド	324
pcat コマンド	340
pdelay コマンド	341
pdisable コマンド	342
pdlink コマンド	343
pdmkdir コマンド	345
pdmode コマンド	346
pdrmdir コマンド	348
pdset コマンド	349
penable コマンド	350
perfwb コマンド	351
pg コマンド	352
phold コマンド	355
pic コマンド	356
pick コマンド	363
ping コマンド	365
pioattred コマンド	370
piobe コマンド	372
pioburst コマンド	374
piocvt コマンド	376
piodigest コマンド	377
piodmgr コマンド	379
piofontin コマンド	380
pioformat コマンド	381
pioquote コマンド	383
piolsvp コマンド	383
piomgpdev コマンド	386
piomkapqd コマンド	387
piomkpq コマンド	389
piomsg コマンド	391
pioout コマンド	393
piopredef コマンド	395
pkgadd コマンド	396
pkgask コマンド	399
pkgchk コマンド	401
pkginfo コマンド	403
pkgmk コマンド	405
pkgparam コマンド	407
pkgproto コマンド	409
pkgrm コマンド	411
pkgtrans コマンド	413
platform_dump コマンド	415
plotgbe コマンド	417

plotlbe コマンド	418
pmctl コマンド	419
pmcycles コマンド	421
pmlist コマンド	422
pmtu コマンド	424
pop3d デーモン	425
pop3ds デーモン	426
portmap デーモン	428
portmir コマンド	429
post コマンド	431
pppattachd デーモン	432
pppcontrold デーモン	435
pppdial コマンド	440
pppstat コマンド	442
pprof コマンド	444
pr コマンド	446
praliases コマンド	448
prctmp コマンド	449
prdaily コマンド	450
preparevsd コマンド	451
preprnode コマンド	452
prev コマンド	454
printenv コマンド	456
printf コマンド	456
probevctrl コマンド	460
probevue コマンド	464
proccred コマンド	469
procfiles コマンド	470
procflags コマンド	472
procldd コマンド	474
procmap コマンド	475
procrun コマンド	479
procsig コマンド	480
procstack コマンド	482
procstop コマンド	483
proctree コマンド	484
procwait コマンド	488
procwdx コマンド	489
prof コマンド	490
proff コマンド	492
projctl コマンド	494
prompter コマンド	503
proto コマンド	505
proxymngr コマンド	506
prs コマンド (SCCS)	508
prtacct コマンド	511
prtconf コマンド	513
prtglbconfig コマンド	517
ps コマンド	518
ps4014 コマンド	540
ps630 コマンド	541
psc または psdit コマンド	542
pshare コマンド	545
psplot コマンド	546
psrasc コマンド	547
psrev コマンド	549

psroff コマンド	549	rdist コマンド	656
pstart コマンド	552	rdistd コマンド	669
pstat コマンド	553	rdump コマンド	670
ptpd デーモン	555	read コマンド	672
ptsc コマンド	561	readlvcopy コマンド	674
ptsevt コマンド	562	reboot または fastboot コマンド	675
ptsevtd コマンド	563	rebootwpar コマンド	676
ptx コマンド	563	recfgct コマンド	677
pvcauth コマンド	565	recreatevg コマンド	679
pvi コマンド	567	recsh コマンド	681
pwchange コマンド	570	redefinevg コマンド	682
pwck コマンド	573	reducevg コマンド	683
pwd コマンド	574	refer コマンド	685
pwdadm コマンド	575	refile コマンド	687
pwdck コマンド	577	refresh コマンド	690
pwtokey コマンド	581	refrsrc コマンド	691
pxed コマンド	584	refsensor コマンド	693
q	587	regcmp コマンド	697
qadm コマンド	587	rembak コマンド	698
qcan コマンド	588	remove コマンド	700
qchk コマンド	590	removevsd コマンド	701
qdaemon コマンド	592	rendev コマンド	702
qhld コマンド	593	renice コマンド	703
qmov コマンド	595	reorgvg コマンド	705
qosadd コマンド	596	repl コマンド	706
qoslist コマンド	598	replacepv コマンド	710
qosmod コマンド	599	repquota コマンド	712
qosremove コマンド	600	reset コマンド	713
qosstat コマンド	601	resetsrc コマンド	714
qpri コマンド	603	resize コマンド	719
qprt コマンド	605	resource_data_input 情報ファイル	720
qstatus コマンド	613	restart-secldapclntd コマンド	724
quiz コマンド	615	restbase コマンド	725
quot コマンド	617	restore コマンド	726
quota コマンド	619	restorevgfiles コマンド	737
quotacheck コマンド	621	restvg コマンド	739
quotaon または quotaoff コマンド	623	restwpar コマンド	742
r	625	restwparfiles コマンド	744
raddbm コマンド	625	resumevsd コマンド	746
radiusctl コマンド	629	rev コマンド	747
ranlib コマンド	629	revnetgroup コマンド	748
raso コマンド	630	rexd デーモン	749
ras_logger コマンド	636	rexc コマンド	750
rbacqry コマンド	637	rexcld デーモン	751
rbactoldif コマンド	642	rgb コマンド	752
rc コマンド	644	ripquery コマンド	753
rc.mobip6 コマンド	644	rksh コマンド	754
rc.powerfail コマンド	645	rlogin コマンド	757
rc.wpars コマンド	648	rlogind デーモン	759
rcp コマンド	648	rm コマンド	761
rcvdist コマンド	653	rmail コマンド	764
rcvpack コマンド	653	rmail コマンド	764
rcvstore コマンド	654	rmauth コマンド	764
rcvttty コマンド	655	rmaudrec コマンド	766
		rmC2admin コマンド	770
		rmCCadmin コマンド	771
		rmcccli 情報ファイル	772

rmccctrl コマンド	777	rmtcpip コマンド	861
rmcdomainstatus コマンド	781	rmts コマンド	862
rmcifscred コマンド	784	rmtun コマンド	863
rmcifsmnt コマンド	785	rmusil コマンド	864
rmclass コマンド	786	rmuser コマンド	865
rmcluster コマンド	787	rmvfs コマンド	867
rmcomg コマンド	788	rmvirprt コマンド	868
rmcondition コマンド	790	rmwpar コマンド	869
rmcondresp コマンド	792	rmyp コマンド	870
rmcosi コマンド	796	rncd コマンド	871
rmddel コマンド	797	rncd-confgen コマンド	872
rmdev コマンド	798	rffbib コマンド	874
rmddir コマンド	801	rolelist コマンド	875
rmddom コマンド	802	roleqry コマンド	876
rmf コマンド	803	rolerpt コマンド	878
rmfilt コマンド	805	rollback コマンド	880
rmfs コマンド	806	route コマンド	881
rmgroup コマンド	807	routed デーモン	886
rmiscsi コマンド	809	rpc.pcnfsd デーモン	889
rmitab コマンド	810	rpcgen コマンド	891
rmkeyserv コマンド	811	rpcinfo コマンド	892
rmlpcmd コマンド	812	rpvstat コマンド	895
rmlv コマンド	814	rrestore コマンド	899
rmlvcopy コマンド	816	Rsh コマンド	903
rmm コマンド	817	rsh または remsh コマンド	905
rmnamsv コマンド	819	rshd デーモン	909
rmnfs コマンド	820	rstatd デーモン	912
rmnfsexp コマンド	821	rsyslogd デーモン	912
rmnfsmnt コマンド	822	rtcd デーモン	915
rmnfsproxy コマンド	823	rtl_enable コマンド	915
rmnotify コマンド	824	runacct コマンド	917
rmpath コマンド	824	runact コマンド	921
rmprtsv コマンド	827	runcat コマンド	925
rmpps コマンド	828	runlpcmd コマンド	925
rmqos コマンド	829	rup コマンド	929
rmque コマンド	830	ruptime コマンド	930
rmquedev コマンド	831	ruser コマンド	931
rmramdisk コマンド	832	rusers コマンド	933
rmresponse コマンド	833	rusersd デーモン	935
rmrole コマンド	836	rvsdrestrict コマンド	935
rmrpdomain コマンド	837	rwall コマンド	937
rmrpnode コマンド	840	rwallld デーモン	938
rmrset コマンド	843	rwho コマンド	938
rmrsrc コマンド	844	rwhod デーモン	940
rmsecattr コマンド	848		
rmsensor コマンド	850	特記事項	943
rmserver コマンド	853	プライバシー・ポリシーに関する考慮事項	945
rmsock コマンド	853	商標	945
rmss コマンド	854		
rmssys コマンド	859	索引	947
rmt コマンド	860		

本書について

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て、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・ライン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、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。

項目	説明
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があるパラメーターを示します。
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。

AIX での大/小文字の区別

AIX® オペレーティング・システムでは、すべてで大/小文字の区別があり、英大文字と小文字が区別されます。例えば、**ls** コマンドを使用するとファイルをリストできます。LS と入力すると、システムはそのコマンドが「not found」(見つからない) と応答します。同様に、**FILEA**、**FiLea**、および **filea** は、同じディレクトリーにある場合でも 3 つの異なるファイル名です。望ましくない処理が実行されないよう、常に正しい大/小文字を使用するようにしてください。

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 のサポート

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 の「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、またいくつか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。UNIX 03 ポータブル・アプリケーション開発する正しい方法を決定するには、UNIX System の Web サイト (<http://www.unix.org>) にある The Open Group の「UNIX 03」仕様を参照してください。

n

以下の AIX コマンドは、文字 *n* から始まります。

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 7.1 は、BIND バージョン 9 のみをサポートします。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 **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** はセキュリティー処理が異なるため、**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

項目	説明
/usr/sbin/named	named デーモンが入っています。
/usr/sbin/named9	named9 デーモンが入っています。
/etc/resolv.conf	ドメイン・ネーム・サービスの使用方法を指定します。
/etc/services	ソケット・サービスの割り当てを定義します。
/usr/samples/tcpip/named.boot	サンプルの named.boot ファイルがその使い方の説明と一緒に入っています。
/usr/samples/tcpip/named.data	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。
/usr/samples/tcpip/hosts.awk	/etc/hosts ファイルを /etc/named.rev ファイルに変換するためのサンプル awk 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。
/usr/samples/tcpip/named.dynamic	動的データベース・セットアップが入っています。

関連資料:

233 ページの『**nslookup** コマンド』

8 ページの『**named9** デーモン』

関連情報:

TCP/IP の **rc.tcpip** ファイル

TCP/IP の **hosts** ファイル・フォーマット

named-checkconf コマンド

目的

名前を指定された構成ファイルの構文検査ツール。

構文

```
named-checkconf [ -v ] [ -j ] [ -t directory ] filename [ -z ]
```

説明

named-checkconf コマンドは、名前を指定された構成ファイルに関して構文はチェックしますが、意味構造はチェックしません。

フラグ

項目	説明
-j	ゾーン・ファイルのロード時にジャーナルがある場合は、それを読み取ります。
-t <i>directory</i>	指定されたディレクトリーに現在のディレクトリーを変更して、それによって構成ファイルに含まれるディレクティブが処理されるようにします。
-v	named-checkconf プログラムのバージョンを表示して終了します。
-z	チェックを行い、 named.conf ファイルにあるマスター・ゾーン・ファイルをロードします。
<i>filename</i>	チェック対象の構成ファイル名を指定します。無指定の場合、デフォルト値は /etc/named となります。

終了状況

項目	説明
0	正常終了を表します。
1	エラーを表します。

関連資料:

『named-checkzone、named-compilezone コマンド』

233 ページの『nslookup コマンド』

241 ページの『nsupdate9 コマンド』

関連情報:

host9 コマンド

dnssec-keygen コマンド

named-checkzone、named-compilezone コマンド

目的

名前を指定された構成ファイルのゾーン・ファイル妥当性検査ツールまたは変換ツール。

構文

```
named-checkzone [ -d ] [ -j ] [ -q ] [ -v ] [ -c class ] [ -f format ] [ -F format ] [ -i mode ] [ -k mode ] [ -m mode ] [ -M mode ] [ -n mode ] [ -o filename ] [ -s style ] [ -S mode ] [ -t directory ] [ -w directory ] [ -D ] [ -W mode ] zonename filename
```

```
named-compilezone [ -d ] [ -j ] [ -q ] [ -v ] [ -c class ] [ -f format ] [ -F format ] [ -i mode ] [ -k mode ] [ -m mode ] [ -n mode ] [ -o filename ] [ -s style ] [ -t directory ] [ -w directory ] [ -D ] [ -W mode ] zonename filename
```

説明

named-checkzone コマンドはゾーン・ファイルの構文と整合性をチェックします。このコマンドは、ゾーンのロード時に **named** デーモンが行うのと同じチェックを行います。そのため、**named-checkzone** コマンドは、ゾーン・ファイルを構成してネーム・サーバーに入れる前にこのゾーン・ファイルをチェックするのに役立ちます。

named-compilezone コマンドは **named-checkzone** コマンドに似ていますが、**named-compilezone** コマンドは、常時、ゾーン内容を指定フォーマットで指定ファイルにダンプします。さらに、**named-compilezone** コマンドは、デフォルトで一層厳しいレベル・チェックを適用します。その理由は、そのダンプ出力が、**named** デーモンによりロードされた実際のゾーン・ファイルとして使用されるからです。チェック・レベルを手動で指定する場合、そのチェック・レベルは、名前を指定された構成ファイルで指定されたチェック・レベル以上の厳しさにする必要があります。

フラグ

項目	説明
-c class	ゾーンのクラスを指定します。無指定の場合は、そのクラスはデフォルトで "IN" となります。
-d	デバッグを使用可能にします。
-D	標準フォーマットでゾーン・ファイルをダンプします。この指定は、 named-compilezone コマンドの場合は必ず有効となります。
-i mode	ロード後のゾーン整合性チェックを行います。mode パラメーターの値は、以下のいずれかになります。
full	MX レコード、SRV レコード、および代行 NS レコードが A または AAAA レコード (ゾーン内またはゾーン外のホスト名) を参照するかどうかをチェックします。また、ゾーン内のグルー・アドレス・レコードが、子により公示されたそのレコードと一致するかどうかをチェックします。
full-sibling	兄弟のグルー・チェックを使用不可にしますが、その他の点では、モード full と同じです。
local	MX レコード、SRV レコード、および代行 NS レコードがゾーン内のホスト名を参照するかどうか、または必須グルーが存在するかどうか (すなわち、ネーム・サーバーが子ゾーンにある場合)だけをチェックします。
local-sibling	兄弟のグルー・チェックを使用不可にしますが、その他の点では、モード local と同じです。
none	チェックを使用不可にします。
-j	ゾーン・ファイルのロード時にジャーナルがある場合は、それを読み取ります。
-f format	ゾーン・ファイルのフォーマットを指定します。指定可能な形式は、"text" (デフォルト) と "raw" です。

項目	説明
-F <i>format</i>	指定された出力ファイルのフォーマットを指定します。指定可能な形式は、"text" (デフォルト) と "raw" です。このフラグは、ゾーン内容をダンプしない限りは、どのような影響も与えません。
-k <i>mode</i>	指定された障害モードにより "check-names" チェックを実行します。指定可能なモードには、"fail"、"warn" (デフォルト)、および "ignore" があります。
-m <i>mode</i>	MX レコードがアドレスであるかどうかを知るために、それらをチェックする必要があるかどうかを指定します。指定可能なモードには、"fail"、"warn" (デフォルト)、および "ignore" があります。
-M <i>mode</i>	MX レコードが CNAME を参照するかどうかをチェックします。指定可能なモードには、"fail"、"warn" (デフォルト)、および "ignore" があります。
-n <i>mode</i>	NS レコードがアドレスであるかどうかを知るために、それらをチェックする必要があるかどうかを指定します。指定可能なモードには、"fail"、"warn" (デフォルト)、および "ignore" があります。
-o <i>filename</i>	<i>filename</i> 値で指定したファイルにゾーン出力を書き込みます。
-q	抑止モードを示します (出口コードのみ)。
-s <i>style</i>	ダンプされたゾーン・ファイルのスタイルを指定します。指定可能なスタイルは、"full" (デフォルト) と "relative" です。"full" フォーマットは、独立したスクリプトによる自動処理に最も適しています。一方、"relative" フォーマットは、full よりも人が読めるフォーマットであるため、人手による編集に向いています。このフラグは、ゾーン内容をダンプしない限りは、どのような影響も与えません。また、出力フォーマットがテキストでない場合は何の意味もありません。
-S <i>mode</i>	SRV レコードが CNAME を参照するかどうかをチェックします。指定可能なモードには、"fail"、"warn" (デフォルト)、および "ignore" があります。
-t <i>directory</i>	<i>directory</i> にディレクトリを変更して、それによって構成ファイルに含まれるディレクティブが処理されるようにします。
-v	named-checkzone コマンドのバージョンを印刷して終了します。
-w <i>directory</i>	現行ディレクトリを <i>directory</i> に変更して、それによってマスター・ファイルの \$INCLUDE ディレクティブの相対ファイル名が機能するようにします。これは、 named.conf ファイルにあるディレクトリ節に似ています。
-W <i>mode</i>	非端末のワイルドカードの有無をチェックするかどうかを指定します。非端末のワイルドカードは、ワイルドカード・マッチング・アルゴリズム (RFC 1034) の解釈に失敗した結果として発生する場合があります。指定可能なモードは、"warn" (デフォルト) と "ignore" です。
<i>zonename</i>	チェック対象のゾーンのドメイン名を指定します。
<i>filename</i>	ゾーン・ファイルの名前を指定します。

終了状況

項目	説明
0	正常終了を表します。
1	エラーを表します。

関連資料:

871 ページの『**rndc** コマンド』

関連情報:

dig コマンド

dnssec-keygen コマンド

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。システム起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・ラインに **-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。

項目	説明
startsrc	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。
stopsrc	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。
refresh	named8 デーモンが /etc/named.conf ファイルの再読み取りを実行します。 refresh 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。
traceson	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。
tracesoff	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。
lssrc	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

項目	説明
-b -cConfFile	代替構成ファイルを指定します。
-dDebugLevel	デバッグ・オプションを提供します。 -d フラグで、 named8 デーモンは、デバッグ情報をデフォルトでファイル /var/tmp/named.run 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。
-pPortNumber	named8 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、 named8 デーモンは /etc/services ファイルに定義されたソケットを listen します。 /etc/services ファイルのエントリは domain で始まります。
-wWorkingDirectory	named8 デーモンの作業ディレクトリを変更します。このオプションは、「ディレクトリ」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。
-tRootDirectory	chroot コマンドを使って、 named8 デーモンの新しいルート・ディレクトリになるディレクトリを指定します。
-q	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。
-r	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。
-f	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。

項目	説明
SIGHUP	named8 デーモンは /etc/named.conf ファイルを再読み取りします。 SIGHUP 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。
SIGILL	統計データを named.stats にダンプします。統計データはファイルに付加されます。
SIGINT	named8 デーモンは、現行データベースをファイル /var/tmp/named_dump.db にダンプします。 ダンプ・ファイルでは、ラベル name error の付いた名前は、負のキャッシュ・エントリを示します。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。 data error のラベルが付いた名前も負のキャッシュ・エントリを示します。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。
SIGUSR1	named8 デーモンはデバッグの実行を開始します。それ以降の SIGUSR1 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 /var/tmp/named.run ファイルに書き込まれます。
SIGUSR2	named8 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・ライン上で使用できます。**-s** フラグは、後に続くサブシステムが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

項目	説明
/usr/sbin/named8	named8 デーモンが入っています。
/usr/sbin/named8-xfer	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。
/etc/named.conf	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、 named8 デーモンの構成を指定します。
/etc/resolv.conf	ドメイン・ネーム・サービスの使用方法を指定します。
/etc/rc.tcpip	システムを再始動するたびにデーモンを初期化します。
/etc/named.pid	プロセス ID を保管します。
/etc/services	ソケット・サービスの割り当てを定義します。
/usr/samples/tcpip/named.conf	サンプルの named.conf ファイルがその使い方の説明と一緒に入っています。
/usr/samples/tcpip/named.data	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。
/usr/samples/tcpip/hosts.awk	/etc/hosts ファイルを /etc/named.data ファイルに変換するためのサンプル awk 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。
/usr/samples/tcpip/addr.awk	/etc/hosts ファイルを /etc/named.rev ファイルに変換するためのサンプル awk 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。

関連資料:

233 ページの『nslookup コマンド』

関連情報:

DOMAIN Local Data

resolv.conf コマンド

TCP/IP デーモン

ネーム・サーバー・レゾリューション

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 [ -4 ] [ -6 ] [ -c config-file ] [ -d debug-level ] [ -f ] [ -g ] [ -n #cpus ] [ -p port ] [ -s ] [-t directory ] [-u user] [ -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、**named9** デーモンはデフォルト構成ファイル `/etc/named.conf` を読み取り、すべての初期データを読み取り、照会を `listen` します。

フラグ

項目	説明
-4	ホスト・マシンが IPv6 に対応していても、IPv4 のみを使用します。 4 オプションと -6 オプションは同時に指定できません。
-6	ホスト・マシンが IPv4 に対応していても、IPv6 のみを使用します。 4 オプションと -6 オプションは同時に指定できません。
-c config-file	デフォルトの <code>/etc/named.conf</code> ではなく、 <code>config-file</code> 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、 <code>config-file</code> 値は絶対パス名にする必要があります。
-d debug-level	デーモンのデバッグ・レベルを <code>debug-level</code> 値にセットします。 named9 デーモンからのデバッグ・トレースは、デバッグ・レベルが増すほど、詳細になります。
-f	フォアグラウンドでサーバーを稼働します。
-g	サーバーをフォアグラウンドで実行し、すべてのログギングを強制的に標準エラー <code>stderr</code> に行います。
-n #cpus	複数 CPU を活用するため、 <code>#cpus</code> ワーカー・スレッドを作成します。無指定の場合、 named9 デーモンは存在する CPU 数を判定し、CPU ごとに 1 つのスレッドを作成します。CPU 数が判定できない場合は、 named9 デーモンは 1 つのワーカー・スレッドを作成します。
-p port	ポート <code>port</code> 上の照会を <code>listen</code> します。指定しないと、デフォルトはポート 53 です。
-s	終了時に標準出力 <code>stdout</code> にメモリー使用統計を書き込みます。
-t directory	コマンド・ライン引数の処理後で構成ファイルの読み取り前に指定したディレクトリーに、現在のディレクトリーを変更します。 警告: -u オプションと一緒にこのオプションを使用する必要があります。その理由は、ルートとして実行中のプロセスの現行ディレクトリーを変更することは、大部分のシステムではセキュリティの機能強化を行わないからです。
-u user	特権操作 (特権ポートで <code>listen</code> するソケットの作成など) の完了後に指定したユーザーに、プロセス・ユーザー ID を設定します。
-v	バージョン番号をレポートし、終了します。
-x cache-file	データを <code>cache-file</code> からデフォルト表示のキャッシュにロードします。

シグナル

ルーチン操作で、このネーム・サーバーを制御するのにシグナルを使用できません。**rndc** コマンドを使用する必要があります。

項目	説明
SIGHUP	サーバーを強制的に再ロードします。
SIGINT 、 SIGTERM	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

項目	説明
/usr/sbin/named9	named9 デーモンが入っています。
/etc/named.conf	デフォルトの構成ファイル。
/etc/named.pid	デフォルトの process-id ファイル。

関連資料:

2 ページの『**named-checkzone**、**named-compilezone** コマンド』

233 ページの『**nslookup** コマンド』

872 ページの『**rndc-confgen** コマンド』

関連情報:

dig コマンド

dnssec-keygen コマンド

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・エンタリーを追加する

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

ネーム・サーバー・エンタリーを削除する

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

すべてのネーム・サーバー・エンタリーを削除する

```
namerslv -X [ -I ]
```

ネーム・サーバー・エンタリーを変更する

namerslv -c *DomainName*

ネーム・サーバー・エントリーを表示する

namerslv -s [**-I** | **-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** ファイルに入っており、このファイルは *FileName* 変数によって指定されたファイルに移動します。

項目	説明
-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 <i>FileName</i>	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 <i>DomainName</i>	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。これは -i <i>IPAddress</i> フラグまたは -n フラグと一緒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。

項目	説明
-E <i>FileName</i>	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。 <code>/etc/resolv.conf</code> ファイルを <i>FileName</i> 変数で指定されたファイルに移動します。
-e	<code>/etc/resolv.conf</code> ファイルを <code>/etc/resolv.conf.sv</code> 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 <i>IPAddress</i>	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 <i>SearchList</i>	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

- 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
```

- `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
```

- 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
```

- 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

項目	説明
<code>/usr/sbin/namerslv</code>	<code>namerslv</code> コマンドが入っています。
<code>/etc/html</code>	

関連情報:

TCP/IP の `resolv.conf` ファイル・フォーマット

`lshosts` コマンド

`mkhosts` コマンド

`tracert` コマンド

TCP/IP デーモン

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 ] ] [-o Options] [ 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

項目	説明
<code>-a</code>	. (ドット) および .. (ドット・ドット) ファイル名をリストします。
<code>-i InNumber</code>	<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。
<code>-o Options</code>	仮想ファイルシステム用のインプリメント固有のオプションのコンマ区切りリストを指定します。
	以下のオプションは、拡張ジャーナル・ファイルシステム (JFS2) に固有です。
<code>-o snapshot=snapNamecheck</code>	コマンドの対象となる内部スナップショットの名前を指定します。スナップショットを所有するファイルシステムがマウントされていなければなりません。
<code>-s</code>	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。

セキュリティー

RBAC ユーザーおよび **Trusted AIX ユーザー**への注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティー」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

ファイルシステム

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

項目	説明
-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

項目	説明
<i>Device</i>	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

項目	説明
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

項目	説明
- 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 <i>hostname</i> 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 <i>interface_index</i>	-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 <i>hostname addr</i>	ハードウェア・アドレス <i>addr</i> で <i>hostname</i> 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
```

関連資料:

『**ndpd-host** デーモン』

17 ページの『**ndpd-router** デーモン』

関連情報:

ifconfig コマンド

autoconf6 コマンド

ndpd-host デーモン

目的

ホストの隣接ディスカバリー・プロトコル (NDP) デーモン。

構文

```
ndpd-host [ -d ] [ -v ] [ -t ] [ -c conffile ] [ -r [ValidLifetime PreferredLifetime] ] [ -g ]
```

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・インターフェース・アドレスを含むデフォルト経路を処理します。ただし、**ndpd-host** コマンドは、ホストで送信された静的なデフォルト経路を上書きすることはありません。デーモンが停止されると、デーモンは、存続期間中に作成された接頭部アドレスおよび経路をクリーンアップします。

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ければなりません。

ndpd-host を使用すると、RFC 4941 に定義されたように一時アドレスを生成できます。デーモンを **tempaddr.conf** ファイル・フォーマットで構成することによって、特定のプレフィックスまたはインターフェースに対する一時アドレス生成を使用可能または使用不可にすることができます。**-r** オプションを使用して、一時アドレスの優先かつ有効な存続期間をデフォルトで設定できます。

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 **100** を介してローカル経路指定を設定します。

フラグ

項目	説明
-c <i>conf</i> file	SEND 構成ファイルを指定します。デフォルトでは、 <code>/etc/ndpd/ndpdh.cnf</code> ファイルが構成ファイルとして使用されます。 SEND オプションを使用可能にするには、 <code>clib.rte</code> ファイルセットと OpenSSL をインストールする必要があります。
-d	デバッグを可能にします (例外条件およびダンプ)。
-g	ndpd-host コマンドが、初期化中にすべての静的グローバル IPv6 アドレスを保持できるようにします。
-r [<i>ValidLifetime PreferredLifetime</i>]	一時アドレス生成を使用可能にします。 -r フラグを使用すると、オプションとして、生成された一時アドレスに対してデフォルトの有効かつ優先の存続期間を指定できます。デフォルトでは、このフラグが無指定の場合は一時アドレスは生成されません。
-t	それぞれのログにタイム・スタンプを追加します。
-v	関心のあるすべてのイベント (<code>daemon.info</code> およびコンソール) をログに記録します。

シグナル

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

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権命令を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権についての詳細情報は、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

ファイル

項目	説明
/etc/ndpd/ndpdh.cnf	SEND ファイルの位置を指定します。
/etc/ndpd/cgaparams.sec	SEND オプションを使用する各インターフェースごとに、その構成を指定します。
/etc/ndpd/sendh_anchor	SEND オプションに必要なトラスト・アンカー値を指定します。
/etc/ndpd/tempaddr.conf	ルーターの接頭部の一時アドレスの生成を拒否または許可するかを指定します。このファイルの内容が読み取られるのは、 <code>-r</code> フラグ指定で <code>ndpd-host</code> が開始される場合に限りです。

関連資料:

881 ページの『route コマンド』

『ndpd-router デーモン』

関連情報:

ifconfig コマンド

ndpdh.cnf コマンド

cgaparams.sec 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 [involife]/[deplife] ] [ -T [reachtim]/[retrans]/[hlim] ] [ -e [ off | compatible | only ] ]
```

説明

`ndpd-router` デーモンは、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。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

項目	説明
-e [off compatible only]	以下のような SEND モードを指定します。 off SEND オプションが使用不可であることを意味します。例えば、ルーターは RFC 3971/3972 より以前のままで機能します。 compatible ルーターは RFC 3971/3972 に準拠しますが、RFC で指定されたオプションは不要です。この環境では、あるノードが SEND 可能であり、他ノードが不可能であるような環境にすることができます。ただし、SEND オプションが着信パケットに組み込まれている場合は、そのオプションは正しい必要があります。 only すべてのメッセージが RFC 3971/3972 に準拠する必要があり、準拠しない場合はそのメッセージが拒否されることを意味します。 SEND オプションを使用可能にするには、 <code>clirc.rte</code> ファイルセットと <code>OpenSSL</code> をインストールする必要があります。
-H	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。
-m	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。
-D <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 です。有効範囲は 1 から 0.75 * <i>max</i> までです。ルーターのライフタイムは <i>life</i> を使用して設定されます。このデフォルト値は 10 * <i>max</i> です。有効範囲は 0 秒から 65535 秒までです。
-T [<i>reachtim</i>] / [<i>retrans</i>] / [<i>hlim</i>]	<i>reachtim</i> がゼロでなければ、 <code>BaseReachableTime</code> フィールドを <i>reachtim</i> 秒に設定します。 <i>retrans</i> がゼロでなければ、 <code>RetransTime</code> 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。
-M	通知の M フラグ (状態構成) を設定します。
-O	通知の O フラグ (その他の状態情報) を設定します。
-P	接頭部 (インターフェース構成から入手) は提供しません。
-P [<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。
-r	ルーター通知でデフォルト・ルーターを提供しません。
-s	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)。
-q	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。
-g	RIPng でデフォルト経路指定をブロードキャストします。
-n	RIPng で受信した経路指定はインストールしません。
-u <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。
-R	<code>split horizon without corrupting reverse</code> を RIPng に使用します。
-S	どのような <code>split horizon</code> (スプリット・ホライズン) も RIPng に使用しません。
-d	デバッグを可能にします (例外条件およびダンプ)。
-v	関心のあるすべてのイベント (<code>daemon.info</code> およびコンソール) をログに記録します。
-t	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*

項目	説明
if = <i>list</i> interface = <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。
mtu [= <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。
ripin =(<i>n y</i>)	着信の RIPng パケットの <i>listen</i> は行いません。 RIPng パケットの送信 (<i>send</i>) は行いません。 <i>split horizon</i> (スプリット・ホライズン) を -S フラグでは使用しません。 -R フラグの場合は、 <i>split horizon without poisoning reverse</i> を使用します。
rtadv =(<i>n y min[/max]</i>)	ルーター通知の送信 (<i>send</i>) は行いません。 <i>min[/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。
flag =[M O]	ルーター通知で状態モード・フラグを設定します。 M 状態構成を使用します。 O 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。
life = <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。
reach = <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。
retrans = <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` が無い場合は、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
```

項目	説明
<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。 prefix ディレクティブには if 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 L または A (あるいはその両方の) フラグを設定します (デフォルトは LA 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=xx::/NumberOfBits gateway=xxx length=(=|>|=|<=|<|>)NumberOfBits
action=(accept|reject|truncate/xx)
```

項目	説明
if =list または interface =list	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、コマンドで区切られます。例えば、 interface =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。
prefix =xx::/NumberOfBits	パターンは、xx::/NumberOfBits 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。
gateway =xxx	パターンは、RIPng メッセージがソース・アドレス xxx 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。
length =(= > = <= < >)NumberOfBits	パターンは、RIPng メッセージ内の接頭部の長さが NumberOfBits 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。
action =(accept reject truncate/NumberOfBits)	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 NumberOfBits 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:

```
xx::/NumberOfBits metric Value
```

```
xx::/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。すべてのフィルターの記述が `cti0` に適用するので、行の順序は重要です。

```
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。

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

ファイル

項目	説明
<code>/etc/gateway6</code>	
<code>/etc/ndpd/sendr_anchor</code>	証明書チェーン用の SEND ルーター・アンカー・ファイル。

関連資料:

644 ページの『`rc.mobip6` コマンド』

15 ページの『`ndpd-host` デーモン』

関連情報:

`ifconfig` コマンド

`autoconf6` コマンド

モバイル IPv6

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・ラインは **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かを伝えます。これらのコマンドは以下を行います。

項目	説明
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

項目	説明
<i>SubjectFile</i>	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
```

主題は 1 桁目から始まらなければなりません。

FormatterCommandLine 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・ラインについて詳しくは、**mmt** コマンド、**nroff** コマンド、および **html** を参照してください。

関連資料:

230 ページの『**nroff** コマンド』

関連情報:

mm コマンド

mmt 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。 **.EN** マクロで始まる行。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。

ターゲットの出力デバイスによっては、 **nroff** コマンドによってフォーマットされた **neqn** コマンドの出力には、 **eqn** コマンドによる後処理が必要になることがあります。このコマンドは、入力フォーマットおよび使用されているキーワードについて、より詳しい情報を提供します。

フラグ

項目	説明
-d <i>Delimiter1Delimiter2</i>	.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、 <i>Delimiter1</i> および <i>Delimiter2</i> 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 Delimiter1Delimiter2 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。区切り文字または .EN マクロに囲まれていないテキストはすべて、未処理のまま渡されます。
-f <i>Font</i>	neqn コマンドで処理したすべてのテキストのフォントを <i>Font</i> 変数で指定した値に変更します。 <i>Font</i> 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 <i>Number</i>	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 <i>Size</i>	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 <i>Size</i> 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

項目	説明
<code>/usr/share/lib/pub/eqnchar</code>	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

`eqn` コマンド

`mm` コマンド

`tbl` コマンド

`.EN` コマンド、`.EQ` コマンド、`mm` コマンド

netcd デーモン

目的

ネットワーク・キャッシング (`netcd`) デーモンを起動します。

構文

```
netcd [ -l file ] [ -c file ] [ -d level ] [ -h ]
```

説明

`netcd` デーモンは、リゾルバーから取得した応答をキャッシングして、ローカル、DNS、NIS、およびユーザー・ロード可能モジュール・サービスが照会に応答する時間を短縮します。

リゾルバー (例えば、DNS) およびマップ (例えば、ホスト) 用に構成されている `netcd` デーモンが実行されると、まずキャッシュされた応答を使用して解決が試みられます。それが失敗した場合は、リゾルバーが呼び出され、応答が `netcd` デーモンによりキャッシュされます。

ローカル、NIS、およびユーザー・ロード可能モジュールの解決用にサポートされているマップのタイプは、ホスト、サービス、ネットワーク、プロトコル、およびネットグループです。DNS の場合、使用できるマップのタイプはホストだけです。

さらに、Yellow Page の特定の場合作のために、以下のマップが追加されています。

- `passwd.byname`
- `passwd.byuid`
- `group.byname`
- `group.bygid`
- `netid.byname`
- `passwd.adjunct.byname`

構成ファイルを使用して、構成が必要なリゾルバーおよびマップを指定できます。このファイルを使用して、他の `netcd` パラメーターも指定できます。デフォルトでは、`/etc/netcd.conf` ファイルが構成ファイルとして使用されます。 `netcd` デーモンの `-c` 引数を使用すると、この構成ファイルのパスを変更できます。 `/etc/netcd.conf` ファイルが存在しない場合、`netcd` デーモンはデフォルト・パラメーターを使用します。 `/usr/samples/tcpip` ファイルにこのファイルのサンプルがあります。このファイルを構成ファイルとして使用しないでください。構成ファイルとして使用すると、このファイルを含むパッケージが新規にインストールされたときに上書きされます。

-d 引数を使用して、デバッグのレベルを指定できます。このデバッグ・レベルは、**syslogd** デーモンで使用されるデバッグ・レベルに類似しています。ログ・メッセージは **/var/tmp/netcd.log** ファイルに書き込まれます。 **netcd** 構成ファイルを使用して、デフォルトをオーバーライドできます。 **syslogd** デーモンの場合と同様に、**netcd** ログ・ファイルの回転を指定できます。

netcd パラメーター

netcd キャッシュにエントリーが挿入されると、存続時間 (TTL) が関連付けられます。この TTL は、**netcd** 構成ファイルを使用して構成できます (キャッシュ宣言)。 DNS の場合、この TTL は、DNS からの応答を含む TTL です。

古くなったエントリーをキャッシュから消去するには、ローカル・キャッシュのクリーニングとその他のキャッシュのクリーニングの 2 つのタスクを定期的に行う必要があります。これらのタスクの頻度は、**netcd** 構成ファイルで **local_scan_frequency** および **net_scan_frequency** パラメーターを使用して設定できます。

キャッシュはハッシュ・テーブルです。ハッシュ・テーブルのサイズは、**netcd** 構成ファイルと **netcdctl** コマンドを使用して制御できます。

netcd デーモンは、アプリケーション間の通信にソケット (**/dev/netcd**) を使用します。 **netcd** 構成ファイルを使用すると、メッセージ・キューのサイズを構成できます。

netcd によるシステム・リソース・コントローラーのサポート

netcd デーモンは、システム・リソース・コントローラー (SRC) グループの一部です。 **netcd** デーモンを管理するために使用できる SRC コマンドは次のとおりです。

- **netcd** を開始するには **startsrc** コマンドを、**netcd** デーモンを停止するには **stopsrc** コマンドを使用できます。
- **lssrc** コマンドは、プロセス ID (PID) および **netcd** デーモンの状況を含む簡略状況出力を提供します。
- **lssrc -l** コマンドは、PID、**netcd** デーモンの状況、**netcd** デーモンの開始時に使用される構成ファイル、および構成済みキャッシュを含む詳細状況出力を提供します。

注: **refresh** コマンドは **netcd** デーモンと一緒に使用できません。

フラグ

項目	説明
-c file	構成ファイルを指定します。デフォルト・ファイル名は /etc/netcd.conf です。
-d level	ロギング・レベルを指定します。 level 値は 0 から 7 までの整数でなければなりません。
-h	ヘルプ情報を表示します。
-l file	netcdctl コマンドで作成された指定のバイナリー・ファイルからキャッシュをロードします。ローカル・ファイル (例えば、 /etc/hosts 、 /etc/services) は、構成ファイルに応じてロードされます。

例

1. SRC を使用して **netcd** デーモンを起動するには、次のように入力します。

```
startsrc -s netcd
```
2. SRC を使用して **netcd** デーモンの状況を表示するには、次のように入力します。

```
lssrc -s netcd
```

このコマンドは、次の出力を生成します。

Subsystem	Group	PID	Status
netcd	netcd	299064	active

3. SRC を使用して **netcd** デーモンの状況を詳細形式で表示するには、次のように入力します。

```
lssrc -l -s netcd
```

このコマンドは、次の出力を生成します。

Subsystem	Group	PID	Status
netcd	netcd	299064	active
Configuration File	/etc/netcd.conf		
Configured Cache	local services		
Configured Cache	local protocols		
Configured Cache	local hosts		
Configured Cache	local networks		
Configured Cache	local netgroup		

4. SRC を使用しないで **netcd** デーモンを起動するには、次のように入力します。

```
netcd
```

関連資料:

『netcdctrl コマンド』

関連情報:

startsrc コマンド

stopsrc コマンド

lssrc コマンド

/etc/netcd.conf コマンド

netcdctrl コマンド

目的

ネットワーク・キャッシング (netcd) デーモンのキャッシュを管理します。

構文

```
netcdctrl [ -t type -e type [ -a file | -b file | -f | -s file ] ] [ -l level ] [ -h ]
```

説明

netcdctrl コマンドは以下の機能を提供します。

- 特定のキャッシュを ASCII フォーマットでダンプする。キャッシュ内容を読みやすい形式で出力します。
- 特定のキャッシュをバイナリー・フォーマットでダンプする。このバイナリー・フォーマットは、後で **netcd** デーモンを起動するときにキャッシュの再ロードに使用できます。このダンプにより、キャッシュを最初から再ロードする必要がなくなります。
- キャッシュ使用状況の統計情報を表示する。キャッシュはテーブルであり、これらのテーブルへのアクセスはハッシュ・アルゴリズムにより制御されます。解像度とマップが与えられている場合、netcd 構成ファイルを使用してテーブルのサイズを決めるときにこの出力を利用できます。
- 特定のキャッシュをフラッシュする。指定されたキャッシュの内容が消去されてから、ローカル・キャッシュが再ロードされます。それ以外のキャッシュは、リゾルバーの応答によって再ロードされます。
- ログギング・レベルを動的に変更する。

要件: **netcdctrl** コマンドを実行するためには、root 権限が必要です。

フラグ

項目	説明
-a <i>file</i>	指定されたキャッシュの ASCII ダンプを指定します。
-b <i>file</i>	指定されたキャッシュのバイナリー・ダンプを指定します (ローカル・キャッシュはダンプされません)。
-e <i>type</i>	マップを指定します。 <i>type</i> パラメーターの値は、以下のいずれかになります。 <ul style="list-style-type: none">• ホスト• protocols• servers• networks• netgroup• Yellow Page マップ名 (例えば、passwd.byname または group.bygid)• all
-f	このフラグは、必ず -b 、 -a 、 -f 、および -s フラグと一緒に使用してください。指定されたキャッシュをフラッシュします。
-h	ヘルプ情報を表示します。
-l <i>level</i>	netcd デーモンのロギング・レベルを変更します。 <i>level</i> 値は 0 から 7 までの整数でなければなりません。
-s <i>file</i>	キャッシュ使用状況の統計情報を提供します。
-t <i>type</i>	解像度を指定します。 <i>type</i> パラメーターの値は、以下のいずれかになります。 <ul style="list-style-type: none">• local• dns• nis• yp• ulm• netcd.conf ファイルにある特定のモジュール名• all

このフラグは、必ず **-b**、**-a**、**-f**、および **-s** フラグと一緒に使用してください。

例

- すべてのキャッシュをフラッシュするには、次のように入力します。

```
netcdctrl -t all -e all -f
```
- すべての NIS キャッシュをバイナリー・フォーマットでダンプするには、次のように入力します。

```
netcdctrl -t nis -e all -b /tmp/netcd_nis_binary_dump
```
- ホストのローカル・キャッシュを ASCII フォーマットでダンプするには、次のように入力します。

```
netcdctrl -t local -e hosts -a /tmp/netcd_dns_hosts
```
- 可能なトレースをすべて取得できるようにロギングのレベルを設定するには、次のように入力します。

```
netcdctrl -l 7
```

関連資料:

25 ページの『**netcd** デーモン』

関連情報:

/etc/netcd.conf コマンド

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 Gensyms_File ] [ -@ [WparList | ALL] ]
```

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せん。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。このモードでは、**gensyms** コマンドによって生成されたファイルも必要です。**gensyms** 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・ライン・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常に、**-o** フラグを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

項目	説明
-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 <i>Trace_File</i> 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 Gensyms_File	gensyms コマンドで生成されたファイル <i>Gensyms_File</i> 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 <i>File</i> 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・デバイス・ドライバーの入出力。このレポートは、オンライン・モードのワークロード区画 (WPAR) 内部または '-@ WparList' フラグを指定されたグローバル WPAR では使用できません。
	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、 -@ フラグを指定せずにグローバル WPAR で netpmon コマンドが実行される場合のデフォルト値です。
-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 <i>n</i> 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、イベントのバーストを調整できます。(通常のイベント・レコード・サイズは 30 バイトくらいです。) <p>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 <i>n</i> 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。</p>
-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。
-@ [WparList ALL]	引数として渡された WPAR のリストにレポートを限定することを指定します。

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・ドライバー・レポート

ネットワーク・デバイス・ドライバー統計情報 _ 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 アクティビティー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
trcoff
gensyms > gen.out
trcstop
netpmon -i tracefile -n gen.out -o netpmon.out
```

関連情報:

trcstop コマンド

gensyms コマンド

recv コマンド

send コマンド

sendto コマンド

netrule コマンド

目的

インターフェースおよびホストについて、規則、フラグ、およびセキュリティー・ラベルの追加、除去、または照会を行います。

構文

```
netrule hl [ i | o | io ]
```

```
netrule hq { i | o } src_host_rule_specification dst_host_rule_specification
```

```
netrule h- [ i | o ][u] [ src_host_rule_specification dst_host_rule_specification ]
```

```
netrule h+ { i | o } [ u ] src_host_rule_specification dst_host_rule_specification [ flags ][ RIPS0/CIPSO options ] security_label_information
```

```
netrule il
```

```
netrule iq interface
```

```
netrule i- [ u ][interface ]
```

```
netrule i+ [ u ] interface [ flags ][ RIPS0/CIPSO options ] security_label_information
```

```
netrule eq
```

```
netrule e { on | off }
```

説明

netrule コマンドは、インターフェースおよびホストの規則仕様のリスト、照会、追加、および除去を行います。システム・デフォルト・インターフェース規則はインターフェース名を使用して設定されます。**i-**フラグを使用してインターフェース規則を除去すると、そのインターフェースにはデフォルト・インターフェース規則が適用されます。デフォルト・インターフェース規則は、**tninit load** コマンドを使用して設定することもできます。

注: インターフェースには必ずインターフェース規則が存在しなければならないので、除去操作を行うとインターフェース規則がデフォルト状態に設定されます。コマンド・ライン・フラグは、すべて構文ステートメントに示す順序で指定する必要があります。

フラグ

項目	説明
e { on off }	システムが受け入れない着信パケットに ICMP エラー応答を送信するポリシーを設定します。この設定はデフォルトではオフであり、オンにするには、このフラグで設定する必要があります。 h フラグまたは i フラグを設定するときは、 e フラグは設定できません。
h	netrule コマンドの対象がホストであることを指定します。 i フラグまたは e フラグを設定するときは、 h フラグは設定できません。
i	netrule コマンドの対象がインターフェースであることを指定します。 h フラグまたは e フラグを設定するときは、 i フラグは設定できません。
l	インターフェースまたはホストに関する規則をすべてリセットします。
o	ホスト発信規則を指定します (ホスト規則の場合のみ)。
q	インターフェース、ホスト規則、またはエラー応答設定の状況を照会します。
u	ホスト規則またはインターフェース規則が正常に追加または除去された後、 /etc/security/rules.host ファイルおよび /etc/security/rules.int ファイルを更新することを指定します。
+	インターフェース規則またはホスト規則を追加します。
-	インターフェース規則またはホスト規則を除去します。
<i>interface</i>	インターフェース名を指定します。
<i>src_host_rule_specification</i>	このパラメーターのフォーマットは次のとおりです。

```
src_host [/ mask][ = proto [:start_port_range [:end_port_range]]]
```

要件: 各フィールド間にスペースまたはタブがあります。

src_host

ソースの IPv6 アドレス、または IPv4 アドレス、またはホスト名。

mask サブネット・マスク番号。最上位ビットを含めて、設定されているビットの数を示します。例えば、IPv4 アドレスの場合、24 は 255.255.255.0 を意味します。

proto プロトコル。

start_port_range

範囲が始まる特定のポート番号またはポート名。

end_port_range

範囲が終わる特定のポート番号またはポート名。

dst_host_rule_specification

このパラメーターのフォーマットは次のとおりです。

```
dst_host [/ mask][ = proto [:start_port_range [:end_port_range]]]
```

要件: 各フィールド間にスペースまたはタブがあります。

dst_host

宛先の IPv6 アドレス、または IPv4 アドレス、またはホスト名。

mask サブネット・マスク番号。最上位ビットを含めて、設定されているビットの数を示します。例えば、IPv4 アドレスの場合、24 は 255.255.255.0 を意味します。

proto プロトコル。

start_port_range

範囲が始まる特定のポート番号またはポート名。

end_port_range

範囲が終わる特定のポート番号またはポート名。

項目
flags

説明

このパラメーターのフォーマットは次のとおりです。

-d drop

drop AIX トラストッド・ネットワークは、すべてのパケットをドロップするように構成できます。以下のいずれかの値を指定できます。

r すべてのパケットをドロップする。

n すべてのパケットをドロップしない (インターフェース・デフォルト)。

i インターフェース・デフォルトを使用する (ホスト・デフォルト、ホストのみ)。

-f rflag:tflag

rflags 着信 (受信) パケットのセキュリティー・オプション要件。以下のいずれかの値を指定できます。

r Revised Interconnection Protocol Security Option (RIPSO) のみ。

c Commercial Internet Protocol Security Option (CIPSO) のみ。

e RIPSO または CIPSO のいずれか。

n RIPSO と CIPSO のいずれも使用しない (システム・デフォルト)。

a 制限なし。

i インターフェース・デフォルトまたはシステム・デフォルトを使用する (デフォルト)。

tflag 発信 (送信) パケットのセキュリティー・オプションの処理。以下のいずれかの値を指定できます。

r RIPSO を送信する。

c CIPSO を送信する。

n セキュリティー・オプションを送信しない (インターフェース・デフォルト)。

i インターフェース・デフォルトを使用する (ホスト・デフォルト、ホストのみ)。

項目
RIPSO/CIPSO オプション

説明
このパラメーターのフォーマットは次のとおりです。

-rpafs=PAF_field[,PAF_field...]

IPSO パケットの受信に使用する PAF フィールドを指定します。これは受け入れられる PAF フィールドのリストです。最大 256 フィールドを指定できます。

PAF_field: NONE | PAF [+PAF...]

PAF のコレクションである PAF フィールドを指定します。1 つの PAF フィールドに次の 5 つの PAF を含めることができます。

- GENSER
- SIOP-ESI
- SCI
- NSA
- DOE

PAF フィールドは、正符号 (+) で区切られたこれらの値の組み合わせです。例えば、GENSER と SCI の両方を含む PAF フィールドは GENSER+SCI と表されます。PAF を 1 つも含まない PAF フィールドを指定するには、PAF フィールド NONE を使用します。

-epaf=PAF_field

システムが受け入れなかった着信 IPSO パケットに対するエラー応答に接続される PAF フィールドを指定します。

-tpaf=PAF_field

発信パケットの IPSO オプションに組み込まれる PAF フィールドを指定します。

-DOI = doi

CIPSO パケットの解釈ドメイン (DOI) を指定します。着信パケットはこの DOI を持つ必要があり、発信パケットはこの DOI を与えられます。

-tags=tag[,tag...]

tag = 1 | 2 | 5

受け入れられて CIPSO オプションによる送信が可能なタグのセットを指定します。1、2、および 5 の組み合わせで指定します。例えば、1,2 と指定すると、タグ 1 と 2 が送信可能になります。

security_label_information

このパラメーターのフォーマットは次のとおりです。

+min +max +default | -s input_file

規則を追加するときに適用される標準出力 (SL) を指定します。また、**-s** フラグを指定して、ファイルに SL を次の順序で含めることができます (1 行に 1 つずつ指定します)。

- min SL
- max SL
- default SL

このファイルにコメントを入れることはできません。複数の行が必要な場合は、行末に円記号 (¥) を使用します。ファイルを使用しない場合は、最小レベル、最大レベル、およびデフォルト・レベル (マークなしパケットの暗黙レベル) の機密ラベルを正符号 (+) で区切ってリストします。

セキュリティ

netrule コマンドを実行するためには、ユーザーは **aix.mls.network.config** 権限と **aix.mls.network.init** 権限を持っている必要があります。

例

1. ホスト着信規則を追加し、ホスト着信規則がカーネルに正常に追加された後でローカル・データベースを更新するには、次のように入力します。

```
netrule h+iu 9.3.149.25 9.41.86.19 +impl_lo +ts all +pub
```

2. ホスト発信規則を追加するには、次のように入力します。

```
netrule h+o 9.41.86.19 9.3.149.25 -s /tmp/rule
```

または:

```
impl_lo
ts all
pub
```

入力 **/tmp/rule** ファイルの内容は次のようになります。

```
impl_lo
ts ¥
all
pub
```

3. ホストからすべての着信 UDP パケットをドロップするには、次のように入力します。

```
netrule h+i 192.0.0.5 =udp 9.41.86.19 =udp -dr +impl_lo +impl_lo +impl_lo
```
4. すべてのホスト規則を除去し、ローカル・データベースを更新するには、次のように入力します。

```
netrule h-u
```
5. すべてのホスト規則をリストするには、次のように入力します。

```
netrule h1
```
6. すべてのインターフェース規則をリストするには、次のように入力します。

```
netrule i1
```
7. インターフェース規則を追加するには、次のように入力します。

```
netrule i+ en0 -dn -fa:n +public +ts +secret
```
8. 特定のホスト規則を除去するには、次のように入力します。

```
netrule h-i 192.0.0.5 =udp 9.41.86.19 =udp
```
9. 特定のホスト規則を追加するには、次のように入力します。

```
netrule h+i 9.41.86.19 /24 =tcp :ftp :telnet 9.3.149.6 /28 +public +ts +secret
```
10. デフォルト・インターフェース規則を設定するには、次のように入力します。

```
netrule i+ default -dn -fa:n +impl_lo +ts all +impl_lo
```
11. デフォルト・インターフェース規則をシステム・デフォルト「すべてのパケットをドロップ」に設定するには、次のように入力します。

```
netrule i- default
```
12. CIPSO パケットを送信し、CIPSO パケットのみを受信するようにインターフェースを設定するには、次のように入力します。

```
netrule i+ en0 -fc:c +impl_lo +ts all +impl_lo
```
13. CIPSO または RIPSO のいずれかを受信し、PAF 値、CIPSO DOI、および CIPSO フラグを持つ RIPSO パケットを送信するようにインターフェースを設定するには、次のように入力します。

```
netrule i+ en0 -fe:r -rpafs=SCI,NSA+DOE -epaf=SCI -tpaf=NSA -DOI=0x010
-tags=1,2 +impl_lo +ts all +impl_lo
```
14. 無効な着信パケットに ICMP 応答を送信する全システム・ポリシーを設定するには、次のように入力します。

netrules on

関連情報:

tninit コマンド

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 } | { -r -C -i -I Interface } ] [ -f AddressFamily ] [ [ -p Protocol ] | [ -@ WparName ] ] [ 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 ] | [ -@ WparName ] ] [ 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

項目

-A

説明

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。このフラグが指定されない場合、インターフェースにバインドされていないサーバー・プロセスによって使用さ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。

項目	説明
-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 <i>AddressFamily</i> 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。 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。
-I Interface	<i>Interface</i> 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・メモリの MBUF クラスタ・プール統計情報を表示します。
-m	メモリ管理ルーチンが記録した統計情報値を表示します。
-n	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、 netstat 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。
-o	-a 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。
-p Protocol	<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。 <i>Protocol</i> 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、 /etc/networks ファイルにリストされます。
-P	Data Link Provider Interface (DLPI) の統計情報を表示します。 netstat -P 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
-r	経路指定テーブルを表示します。-s フラグと一緒に使用すると、-r 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。
-s	各プロトコルの統計情報を表示します。
-ss	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。
-u	ドメイン・ソケットに関する情報を表示します。
-v	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、 netstat コマンドが、 netstat 、 tokstat 、および fdldistat 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・ドライバー・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・ドライバー統計情報コマンドのセクションを参照してください。

項目	説明
-Zc	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。
-Zi	インターフェース統計情報をクリアします。
-Zm	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。
-Zs	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、 -p <protocol> を使用します。例えば、TCP 統計情報をクリアする場合は、 netstat -Zs -p tcp と入力します。
-@ WparName	ワークロード区画 (WparName) に関連付けられたネットワーク統計情報を表示します。 WparName を指定しない場合は、すべてのワークロード区画のネットワーク統計情報を表示します。

注:

1. **-C**、**-D**、**-c**、**-g**、**-m**、**-M**、**-P**、**-r**、**-v**、および **-Z** フラグは、**-@ WparName** オプションと一緒に使用される場合、グローバル環境でサポートされません。
2. **-C**、**-D**、**-c**、**-g**、**-m**、**-M**、**-P**、**-r**、**-v**、および **-Z** フラグは、システム・ワークロード・パーティションではサポートされません。

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。

ルーティング・テーブルには以下のフィールドが含まれています。

項目	説明
WPAR	この経路が所属するワークロード・パーティションの名前を表示します。このフィールドが存在するのは、 <code>-@</code> フラグが <code>-r</code> フラグと一緒に使用されている場合だけです。グローバル・システムに所属する経路の場合、この列には <code>Global</code> と表示されます。
フラグ	経路指定テーブルの flags フィールドは経路の状態を示します。 A 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。 U 立ち上がっています。 H 経路はネットワークアドレスではなくホスト宛です。 G 経路はゲートウェイ宛です。 D 経路はリダイレクトによって動的に作成されています。 M 経路はリダイレクトによって修正されています。 L リンク・レベルのアドレスは経路エントリーにあります。 c この経路へのアクセスは、クローン経路を作成します。 W この経路はクローン経路です。 1 プロトコル特定の経路指定フラグ #1。 2 プロトコル特定の経路指定フラグ #2。 3 プロトコル特定の経路指定フラグ #3。 b 経路はブロードキャスト・アドレスを示します。 e 割り当てキャッシュ・エントリーを使用します。 l 経路はローカル・アドレスを示します。 m 経路はマルチキャスト・アドレスを示します。 P ピン固定経路。 R 到達不可能なホストまたはネットワーク。 S 手動で追加。 u 使用可能な経路。 s 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 gateway 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。コネクション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。

項目	説明
Route Tree for Protocol Family	<p>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とおりです。</p> <p>1 UNIX アドレス・ファミリーを指定します。</p> <p>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)</p> <p>他のアドレス・ファミリーの詳細は、<code>/usr/include/sys/socket.h</code> ファイルを参照してください。</p>

-@ フラグが **netstat -r** コマンドで使用され、かつ、*WparName* パラメーターが無指定の場合、このシステムの経路テーブルにある経路すべてが表示されます。*WparName* パラメーターが指定され、かつ、WPAR 固有の経路指定がその WPAR に対して使用可能な場合、その WPAR に関連する経路のみが表示されます。*WparName* パラメーターが指定され、かつ、WPAR 固有の経路がその WPAR に対して使用不可の場合、そのグローバル・システムに関連する経路が表示されます。

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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**Issecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

- 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、以下のコマンドを入力します。

```
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
(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。

- GRE プロトコルの統計情報を表示するには、以下のコマンドを入力します。

```
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
```

- IPV6 トンネル (GIF トンネル) 経由の IPV4 に関する統計を表示するには、以下のコマンドを入力します。

```
netstat -s -p gif
```

このコマンドは、次の出力を生成します。

```
GIF Interface gif0
 44 total packets received
 50 total packets sent
   0 packets received with protocol not supported
   0 packets received with checksum on
   0 packets received with routing present
   0 packets received with strict source route present
   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 ipv6 output routine
```

- 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、以下のコマンドを入力します。

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

以下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。

5. 各プロトコルの統計情報を表示するには、以下のコマンドを入力します。

```
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。

6. デバイス・ドライバの統計情報を表示するには、以下のコマンドを入力します。

```
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 **fddistat** コマンドのセクションを参照してください。

7. 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、以下のコマンドを入力します。

```
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・インターフェースを示し、**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。

8. 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、以下のコマンドを入力します。

```
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)

9. アクティブなソケットの詳細データを表示するには、以下のコマンドを入力します。

```
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)
    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

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
```

.....

10. 経路指定テーブルを表示するには、次のコマンドを入力します。

```
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。

行末の文字 => は、その行が次の行の経路と重複した経路であることを示します。

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。

11. wpar1 という名前のワークロード・パーティションの経路指定テーブルを表示するには、次のコマンドを入力します。

```
netstat -rn@ wpar1
```

以下のような出力が表示されます。

```

Routing tables
WPAR Destination      Gateway      Flags      Refs      Use      If      Exp      Groups

Route Tree for Protocol Family 2 (Internet):
wpar1 default          9.4.150.1    UG         1      13936   en1      -      -
wpar1 9.4.150.0         9.4.150.57  UHSb       0         0     en1      -      - =>
wpar1 9.4.150/24       9.4.150.57  U          0         0     en0      -      -
wpar1 9.4.150.57       127.0.0.1   UGHS       0         0     lo0      -      -
wpar1 9.4.150.255      9.4.150.57  UHSb       0         3     en0      -      -

```

関連情報:

TCP/IP の protocols ファイル・フォーマット

tokstat コマンド

vmstat コマンド

naming コマンド

TCP/IP アドレッシング

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

項目	説明
0	正常に終了しました。
>0	エラーが発生しました。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

ファイル

項目	説明
/usr/sbin/newaliases	newaliases コマンドが入っています。
/etc/mail/html	

関連情報:

sendmail コマンド

メール別名

別名データベースの構築

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・ライン・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・ライン・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・ライン・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・ライン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

項目	説明
-a [<i>Number</i>]	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 [<i>Character</i>] および -p [<i>Number</i>] フラグも参照してください。
-b [<i>Number</i>]	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 <i>Number</i> 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 [<i>Number</i>] フラグも参照してください。 <i>Number</i> 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。 <code>newform -l1-b7 file-name</code> -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 [<i>Character</i>]	接頭部/追加文字を <i>Character</i> 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 [<i>Number</i>]	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 [<i>Number</i>] 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 を含みます。
-i [<i>TabSpec</i>]	入力のすべてのタブを <i>TabSpec</i> 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。 <i>TabSpec</i> 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。 <i>TabSpec</i> のデフォルト値は -8 です。 <i>TabSpec</i> 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 [<i>Number</i>]	有効行長を指定文字数に設定します。 <i>Number</i> 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。

項目	説明
<code>-o [TabSpec]</code>	タブ指定に従って入力スペースを出力でタブに置換します。 <i>TabSpec</i> のデフォルト値は <code>-8</code> です。 <i>TabSpec</i> の値が <code>-0</code> 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。
<code>-p [Number]</code>	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。 <code>-c</code> フラグも参照してください。
<code>-s</code>	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

項目	説明
-	環境を新グループのログイン環境に変更します。
-i	- 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。

項目	説明
>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

項目	説明
<code>/etc/passwd</code>	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

`/etc/group` ファイル

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・プロシージャ・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。

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することができます。

フラグ

項目	説明
<code>-h HostName</code>	<code>HostName</code> パラメーターで指定されたコンピューターで、 <code>root</code> 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 <code>root</code> 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。
<code>-u UserName</code>	<code>UserName</code> 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

項目	説明
<code>/etc/publickey</code>	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

System Management Interface Tool

ネットワーク・ファイルシステム (NFS) の概要

Network Information Service (NIS)

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

項目	説明
-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 (| 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

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

関連資料:

352 ページの『pg コマンド』

関連情報:

/etc/security/environ コマンド

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) フォルダーの次のメッセージ・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

項目	説明
+Folder	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。
-header	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。
-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。
-noheader	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。 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。
-noshowproc	リスト作成を行うため /usr/bin/cat ファイルを使用します。これはデフォルトです。
-showproc CommandString	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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

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

関連資料:

454 ページの『`prev` コマンド』

関連情報:

`show` コマンド

`.mh_alias` コマンド

`.mh_profile` コマンド

メール・アプリケーション

`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 または 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

項目	説明
-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 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

項目	説明
0	コマンドは正常に完了しました。
1	引数エラー。

例

- 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`
- 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`
- 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。

項目	説明
rw	このパス (fsid) 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。
ro	このパス (fsid) 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。
acdirmax	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。
acdirmin	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。
acregmax	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。
acregmin	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。
cio	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 open() システム・コール内で指定された O_CIO でオープンされたかのように動作します。
cior	読み取り専用ファイルがファイルシステムでオープンできることを指定します。このファイルシステムにあるファイルの入出力は、そのファイルが open() システム・コール内に指定された O_CIO O_CIOR でオープンされたかのように動作します。
dio	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 open() システム・コールで指定された O_DIRECT 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。
hard	この fsid 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。
intr	fsid 操作が割り込み可能であることを指定します。
maxpout=value	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。 maxpout を指定した場合は、 minpout 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、 minpout より大でなければなりません。デフォルトは、カーネル maxpout レベルです。
minpout=value	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。 minpout を指定した場合は、 maxpout 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 minpout レベルです。
noac	属性キャッシュを使用しません。
nocto	クローズとオープン間の整合性を指定しません。
nointr	fsid 中断不可能であることを指定します。
prefer=servername	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、 nfs4cl 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。
rbr	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ーにコピーされると、ファイルに使用されたこのページが解放されます。
rsize	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。
retrans	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。
soft	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。
timeo	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。
wsize	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。
nodirCACHE	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

項目	説明
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`

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

項目	説明
-r <i>hostname</i>	<i>hostname</i> パラメーターまたは <i>IP_address</i> 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。
<i>IP_address</i>	

ファイル

項目	説明
<code>/usr/sbin/nfs4smctl</code>	<code>nfs4smctl</code> コマンドの位置。

関連資料:

65 ページの『`nfs4cl` コマンド』

nfsauthreset コマンド

目的

カーネルのクレデンシャル・キャッシュから該当する Generic Security Service API (GSSAPI) クレデンシャルを廃棄するように、ネットワーク・ファイルシステム (NFS) カーネル拡張子に通知します。

構文

nfsauthreset

説明

nfsauthreset コマンドは、プロセス認証グループ (PAG) がプロセスで設定されているかどうかに応じて、キャッシュされたコンテキストにマークを付けます。プロセスで PAG が設定されている場合は、同じユーザー ID (UID) と PAG を持つキャッシュされた GSSAPI コンテキストに破棄のマークを付けます。PAG が設定されていない場合は、同じ UID を持つキャッシュされた GSSAPI コンテキストに破棄のマークを付けます。

例

kinit コマンドおよび **kdestroy** コマンドを指定した後で、キャッシュされたカーネル・クレデンシャルを破棄するには、次のように入力します。

```
nfsauthreset
```

ファイル

項目	説明
<code>/usr/sbin/nfsauthreset</code>	<code>nfsauthreset</code> コマンドが入っています。

関連情報:

`kinit` コマンド

`kdestroy` コマンド

`env` 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

項目	説明
-a	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。
-c max_connections	NFS サーバーで許可される TCP 接続の最大数を指定します。
-gp on off	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。-gp 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。
-gpbypass	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、-gp 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。
-gpx count	NFSv4 猶予期間の自動延長を制御します。count 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。-gpx 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の期間の範囲を限度に猶予期間を延長します。
nserver	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。chnfs、chssys、または nfso 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。
-p tcp または -p udp	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、-p tcp 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。
-r max_read_size	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容値は 64K です。
-w max_write_size	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容値は 64K です。
-root directory	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 / 以外のもので設定されている場合は、chnfs -r 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 nfsd が実行中ですが、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 chnfs -r を使用してください。
-public directory	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 nfsd が実行中の時に使用できません。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 chnfs -p を使用してください。
-getnodes	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。
-getreplicas	現在の複製使用可能化モードを表示します。レプリカが nfsroot 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。

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

mountd コマンド

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

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

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

1. 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. 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
```

nfshostmap コマンド

目的

nfs 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、マップされることになります。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

項目	説明
-a <i>hostname alias1 alias2</i>	別名からのマッピングを <i>hostname</i> に追加します。
-d <i>hostname</i>	<i>hostname</i> のすべての別名を削除します。
-e <i>hostname alias1 alias2</i>	<i>hostname</i> 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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**Issecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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 [-F] [ -c ]  
nfso -h [ Tunable ]  
nfso -l [ hostname ]  
nfso [-F] -L [ Tunable ]  
nfso [-F] -x [ Tunable ]  
nfso [-@ WparName ] [ -p | -r ] -a [ -c ]  
nfso [-@ WparName ] [ -p | -r ] [ -c ] { -o Tunable[ =newvalue ] }  
nfso -H {ha operation}
```

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。パラメーターの値を表示するかパラメーターに新しい値を設定するか、いずれかを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。

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 の開発側から具体的な指示がない限り、このパラメーターを変更しないようにしてください。

フラグ

項目	説明
-a	すべてのチューナブル・パラメーターについて、現行値とリポート値 (-r を使用した場合) または永続値 (-p を使用した場合) を表示します。それぞれの組み合わせが 1 行あたり 1 つずつ表示されます (<i>Tunable = Value</i>)。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の区切るフォーマットに変更します。
-d Tunable	<i>Tunable</i> 変数をデフォルト値に戻します。 <i>Tunable</i> 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、変更は行われず、警告が表示されます。
-D	すべての <i>Tunable</i> 変数をデフォルト値に戻します。変更する必要がある <i>Tunable</i> 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、変更は行われず、警告が表示されます。
-F	コマンド・ラインでオプション -a 、 -L 、または -x を指定したとき、制限付きチューナブル・パラメーターの表示を強制します。 -F フラグを指定しなかった場合は、表示オプションとの関連で名指して指定されていない限り、制限付きチューナブル・パラメーターは含まれません。
-h [Tunable]	<i>Tunable</i> 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。
-H {ha operation}	ハイ・アベイラビリティ (HA) 操作を実行します。HA 操作は、次のとおりです。
	enable_ha HA 機能をオンにします。
	disable_ha HA 機能をオフにします。
	sm_register <hostname> PowerHA® SystemMirror® はこのホストを登録します。
	sm_unregister <hostname> PowerHA SystemMirror はこのホストを登録抹消します。
	sm_gethost PowerHA SystemMirror はホストを取得します。
	dump_dupcache <log device> HA dupcache をダンプします。
-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。 <i>hostname</i> 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・コールを実行して <i>hostname</i> 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。
	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。

項目

-L [*Tunable*]

説明

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
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

-o *Tunable* [=*newvalue*]

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、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

-@ *WparName*

指定されたワークロード区画のチューナブル・パラメーターを設定または表示します。 **-@** フラグは、**nfso** コマンドがグローバル区画で実行される場合にのみ使用できます。

制限付きチューナブル・パラメーターを変更すると (**-o**、**-d**、または **-D** を指定)、限定使用タイプのチューナブル・パラメーターが変更されたという警告メッセージが表示されます。コマンド・ラインで **-r** または **-p** オプションも指定した場合は、変更の確認を求めるプロンプトが出されます。さらに、システム・リブート時に制限付きチューナブル・パラメーターが */etc/tunables/nextboot* ファイルにあって、それらの

パラメーターがデフォルト値と異なる値に (コマンド・ラインで **-r** または **-p** オプションを指定して) 既に変更されていた場合は、それらの変更されたチューナブル・パラメーターのリストを示すエラー・ログ・エントリーが作成されます。

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** のパラメーターを変更すると (**-o**、**-d**、または **-D** フラグを指定)、**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という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すると (**-o**、**-d**、または **-D** を指定)、エラー・メッセージが出されます。

タイプ **Incremental** のパラメーターの現在の値を、現在の値より小さい新しい値に変更すると (**-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。

注: システム全体に適用されるチューナブル変数は、**workload partition**内では変更できません。

注: **nfso** コマンドをワークロード区画内で実行する場合 (または **-@** フラグが指定されている場合)、**-o** フラグでは次のチューナブル・パラメーターのみを設定できます。

- **nfso_dynamic_retrans**
- **nfso_iopace_pages**
- **nfso_use_reserved_port**
- **nfso_v4_fail_over_timeout**
- **utf8_validation**
- **nfso_auth_rbr_trigger**
- **client_delegation**

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。

項目	説明
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、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)、リブート値は、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、ブート時にリブ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ブート値の設定は、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、パラメーターのリブート値は、リブート中に `/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、「*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

チューナブルのデフォルト値および値の範囲については、`nfso` コマンド・ヘルプ (`-h <チューナブル・パラメーター名 >`) を参照してください。

注: 以下のパラメーターは、AIX バージョン 6.1 (6100-02 テクノロジー・レベル適用) 以降のことになりますが、もう使われなくなります。その理由は、ネットワーク・ファイルシステム (NFS) と 仮想メモリーマネージャー (VMM) が `buf` 構造数とページング・デバイス・テーブル数 (PDT) を、ワークロードに基づいて自動的にチューニングするからです。

- `nfs_v2_pdots`
- `nfs_v2_vm_bufs`
- `nfs_v3_pdots`
- `nfs_v3_vm_bufs`
- `nfs_v4_pdots`
- `nfs_v4_vm_bufs`
- `nfs_server_close_delay`

項目	説明
<code>client_delegation</code>	<p>目的: NFS バージョン 4 クライアントがオープン・ファイルに対する代行を受け入れるかどうかを判別します。</p> <p>チューニング: 値が 0 の場合は、代行が使用不可になります。値が 1 の場合は、代行が使用可能になります。</p>

項目	説明
nfs_hang_log	<p>目的: ハングしている NFS マウント・メッセージを syslog ログ・ファイルに記録させる優先順位を設定します。</p> <p>チューニング: 値は 1 から 7 までの範囲です。</p> <ul style="list-style-type: none"> • 1: LOG_ALERT • 2: LOG_CRIT • 3: LOG_ERR • 4: LOG_WARNING • 5: LOG_NOTICE • 6: LOG_INFO • 7: LOG_DEBUG <p>デフォルト値は 6 です。</p>
nfs_max_read_size	<p>目的: システム管理者がサーバーで NFS RPC サイズを制御できるようにします。</p> <p>チューニング: すべてのクライアントが読み取りサイズを変更する必要があり、かつクライアントを変更できない場合に役立ちます。クライアント・マウントの値をデフォルト値として使用します。クライアントでマウントを直接操作できないとき (特に、デフォルトの読み取りサイズでパケットをドロップしているネットワークでの NIM インストール時に)、V3 読み取りサイズを縮小するためにデフォルト値が必要です。このような場合、512 KB の最大サイズをより小さい値に設定して、その値がネットワーク上で役立つようにします。このパラメーターは、ネットワーク・デバイスがパケットをドロップしており、サーバーとの通信について一般的な変更が要求される場合にも役立ちます。デフォルト値は 64 KB で、最大値は 512 KB です。</p>
nfs_max_write_size	<p>目的: システム管理者がサーバーで NFS RPC サイズを制御できるようにします。</p> <p>チューニング: すべてのクライアントが書き込みサイズを変更する必要があり、かつクライアントを変更できない場合に役立ちます。クライアント・マウントの値をデフォルト値として使用します。クライアントでマウントを直接操作できないとき (特に、デフォルトの読み取りサイズでパケットをドロップしているネットワークでの NIM インストール時に)、V3 書き込みサイズを縮小するためにデフォルト値が必要です。このような場合、512 KB の最大サイズをより小さい値に設定して、その値がネットワーク上で役立つようにします。このパラメーターは、ネットワーク・デバイスがパケットをドロップしており、サーバーとの通信について一般的な変更が要求される場合にも役立ちます。デフォルト値は 64 KB で、最大値は 512 KB です。</p>
nfs_rfc1323	<p>目的: 非常に大きい 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えるようにします。</p> <p>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している場合、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できるようにします。これにより、クライアントとサーバー間で使用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 のみに影響し、システム内の他のアプリケーションには影響しません。値が 0 の場合はこれが使用不可になり、値が 1 の場合はこれが使用可能になります。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は、この NFS オプションを設定する必要はありません。</p>

項目	説明
nfs_securenfs_authtimeout	<p>目的: DES クレデンシャルの秒数を設定します。</p> <p>チューニング: 値が 0 の場合は、DES クレデンシャルのタイムアウトが使用不可になります。</p>
nfs_server_base_priority	<p>目的: nfsd デーモンの基本優先順位を設定します。</p> <p>チューニング: デフォルトでは、nfsd デーモンは浮動するプロセス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更されます。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。許容範囲内の他の値は、NFS 要求をサーバーで受信したときの nfsd デーモンの優先順位を設定するために使用されます。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 (下げるか、または nfsd デーモンの優先順位を低くする)。また、nfsd デーモンをサーバー上で最も優先させるプロセスの 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 サーバーが非常にビジー状態であり、サーバー上でその他のプロセスが実行時間を持っていないように実質的に締め出す場合に起こります。</p>
nfs_server_cread	<p>目的: このオプションを使用すると、NFS サーバーは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なります。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。ただし、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」を実行します。</p> <p>チューニング: これはサーバーのメモリーが低く、ディスクからメモリーへのアクティビティが多く行われている場合に有効です。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。値が 1 の場合は、積極的な先読みが行われます。値が 0 の場合は、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 (JFS ファイルシステムの場合) および JFS2 (JFS2 ファイルシステムの場合) により制御されます。nfs_server_cread オプションで可能になる、より積極的なトップ・ハーフの先読みは、順序に従わない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、一度に 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。</p>
nfs_server_close_delay	<p>目的 予期される遅延があまり長くない場合に、NFS バージョン 4 サーバーは NFS4ERR_DELAY 応答を送信するのを回避する必要があります。サーバーからの NFS4ERR_DELAY 応答を検出したときにアプリケーションを長時間休止する NFS クライアントが使用される場合、サーバーは、アプリケーションの休止を回避する nfs_server_close_delay オプションを使用してサーバー上の遅延を処理しようとしています。</p> <p>チューニング 値が 0 の場合、この機能はオフにされます。デフォルト値は 0 です。値が 1 の場合、サーバー・サイドでの短い遅延をローカル処理することができます。</p>
nfs_use_reserved_ports	<p>目的: 予約されていない IP ポート番号の使用を指定します。</p> <p>チューニング: 値が 0 の場合は、NFS クライアントが NFS サーバーと通信するときに予約されていない IP ポート番号を使用します。</p>

項目	説明
nfs_v3_server_readdirplus	<p>目的: READDIRPLUS 呼び出しがサーバーでサポートされているかどうかを判別します。</p> <p>チューニング: 値が 0 の場合は、READDIRPLUS の処理が使用不可になります。</p>
nfs_v4_fail_over_timeout	<p>目的: NFS バージョン 4 クライアントの操作が、NFS バージョン 4 サーバーが提供するレプリカにフェイルオーバーするタイムアウトの時間を指定します。単位は秒です。</p> <p>チューニング: 値 0 が指定されている場合、タイムアウト値は tcp のタイムアウト値の 4 倍となります。1 から 4 までの値は予約されており、NFS バージョン 4 クライアントはこれを 0 として扱います。NFS バージョン 4 では、メイン・サーバーが応答しない場合、クライアントは他のレプリカ・サーバーにフェイルオーバーすることができます。この値により、クライアントが fsid に対するすべての NFS バージョン 4 の要求を他のレプリカ・サーバーに切り替えるまで、サーバーからの応答を待機する時間が決まります。</p>
portcheck	<p>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。</p> <p>チューニング: 値が 0 の場合は、NFS サーバーが行うポート検査が使用不可になります。値が 1 の場合は、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、最低限のパフォーマンス結果を伴う構成決定です。</p>
server_delegation	<p>目的: NFS バージョン 4 サーバーがオープン・ファイルに対する読み取りの代行を実行するかどうかを判別します。</p> <p>チューニング: 値が 0 の場合は、代行の許可が使用不可になります。値が 1 の場合は、代行の許可が使用可能になります。</p>
utf8_validation	<p>目的: NFS バージョン 4 のクライアントとサーバーがストリング・データで UTF-8 の正確性を検査するかどうかを判別します。</p> <p>チューニング: 値が 0 の場合は、UTF-8 の検査が使用不可になります。値が 1 の場合は、UTF-8 の検査が使用可能になります。</p>

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。
nfsd -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。
nfsd -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、コロンで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。
nfsd -a -c
4. **nfsd** 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。
nfsd -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。
nfsd -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。

```
nfsd -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。

```
nfsd -r -a
```

8. **nfsd** 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。

```
nfsd -x
```

関連資料:

44 ページの『netstat コマンド』

203 ページの『no コマンド』

関連情報:

TCP/IP プロトコル

NFS 統計情報とチューニング・パラメーター

カーネル・チューニング

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・ラインで、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、**chnfsdom** 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

項目	説明
-f	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。
-T	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

項目	説明
<code>/etc/environment</code>	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

`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・プロシージャ・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 [ -@ WparName ] [ -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・プロシージャ・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。

項目	説明
calls	受信した RPC 呼び出しの合計回数。-4 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。
badcalls	RPC レイヤーがリジェクトした呼び出しの合計回数。-4 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。
nullrecv	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。
badlen	最小サイズの RPC 呼び出しより短い RPC 呼び出しの回数。
xdrCALL	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。
dupchecks	重複する要求キャッシュで検索した RPC 呼び出しの回数。
dupreqs	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

項目	説明
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。

項目	説明
auth	次の値のいずれかを指定します。 none 認証なし。 unix UNIX スタイル認証 (UID、GID)。 des 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・サイズをバイト単位で読み取ります。
wsiz	バッファ・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。

項目	説明
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。
動的	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	readdirplus のかわりに readdir 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。

項目	説明
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

項目	説明
-@ WparName	指定されたワークロード区画の統計情報を表示します。-@ フラグは、 nfsstat コマンドがグローバル区画で実行される場合にのみ使用できます。 nfsstat コマンドがワークロード区画から実行されたときに -@ フラグが指定されていない場合は、現行ワークロード区画の統計情報が表示されます。 nfsstat コマンドがグローバル区画から実行されたときに -@ フラグが指定されていない場合は、すべてのアクティブなワークロード区画 (およびグローバル区画) の合計統計情報が表示されます。 注: -m フラグと一緒に -@ WparName フラグを使用すると、 nfsstat コマンドは指定のワークロード・パーティションではなく、グローバル・パーティション用の統計を表示します。
-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。 注: nfsstat コマンドの使用時に、 -m オプションを指定すると、グローバル・パーティション用の統計が常時出力されます。
-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。このオプションが指定されていない場合の出力は、バージョン 5.3 より前のバージョンの AIX における nfsstat コマンド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. 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。

7. ワークロード区画 **abc** の NFS クライアント統計情報を表示するには、次のように入力します。

```
nfsstat -@ abc -cn
```

関連情報:

ネットワーク・ファイルシステム (NFS) の概要

NFS コマンドのリスト

NFS パフォーマンス

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。 **-i** 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: **cs**h 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、**cs**h 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。 **cs**h コマンドの **nice** コマンドについては、**cs**h コマンドを参照してください。

フラグ

項目	説明
-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。

項目	説明
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管理」のシェルを参照してください。

ファイル

項目	説明
<code>/usr/bin/nice</code>	nice コマンドが入っています。

関連資料:

227 ページの『**nohup** コマンド』

703 ページの『**renice** コマンド』

関連情報:

cs コマンド

shells コマンド

マイクロプロセッサのコンテンションの制御

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

項目	説明
-a <i>Attribute = Value . . .</i>	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。
-o <i>Operation</i>	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。
	activate 管理対象のシステムを開始します。
	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 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・レベルにアップグレードする。
	bos_inst BOS インストールを実行します。
変更	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。
	chwpar 管理対象のワークロード・パーティションの特性を変更します。
	create 管理対象システムのインスタンスを作成します。
	cust ソフトウェアのカスタマイズを実行します。
	deactivate 管理対象のシステムを停止します。
	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。
	destroy 管理対象システムのインスタンスを除去します。
	diag システムが診断イメージをブートできるようにします。
	dkls_init システムのディスクレス環境を初期化します。
	dtls_init システムのデータレス環境を初期化します。
	fix_query 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 オペレーティング・システム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 システムおよび 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。
	lslpp 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。
	lswpar 管理対象のワークロード・パーティションの特性を表示します。
	maint ソフトウェアの保守を行います。
	maint_boot システム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・システムをリブートします。

項目
-o Operation (続き)

項目	説明
削除	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。
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。
syncwpar	管理対象のワークロード・パーティション・ソフトウェアと管理元システムとを同期化します。
takeover	alternate_master として構成されているマシンが NIM 環境を制御できるようにします。
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。
updateios	vios または ivm タイプである、仮想入力サーバー (VIOS) 管理サーバー上でソフトウェアのカスタマイズおよび保守を実行します。
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せん。

項目
-t Type

説明
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されるファイルを構成します。

devexports
ワークロード・パーティション用にこのファイルをデバイス・エクスポートします。

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。

fb_script
マシンの最初のリブ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されるファイルを構成します。

installp_bundle
Installp バンドル・ファイル。

ios_mksysb
vios または **ivm** タイプである VIOS 管理サーバーから取られたバックアップ・イメージを表します。

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。

nas_filer network-attached storage (NAS) デバイス。

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

ページング
クライアントのページング・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。

savewpar
ワークロード・パーティションのバックアップ・イメージ。

script
クライアントで実行する実行可能ファイル。

secattr
ワークロード・パーティション用のセキュリティー属性ファイル。

項目
-t Type (続き)

説明
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。

shared_home
クライアント間で共用する /home ディレクトリー。

shared_root
クライアントで共有する / (root) 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。

wpar_spec
ワークロード・パーティション作成用の仕様ファイル。

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システム。

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・ページング、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システム。

wpar 管理元システムがホストするワークロード・パーティション。

管理タイプ:

bcm ブレード管理モジュール・ハードウェア。

cec 集中電子コンプレックス・ハードウェア。

hmc ハードウェア管理コンソール・システム。

ivm 統合仮想管理システム。

vios 仮想入出力サーバー。

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。

generic その他の TCP/IP ネットワーク。

hfi ホスト・ファブリック・インターフェース (HFI) 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。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティー」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

次に示す例は、操作ごとにまとめてあります。

activate

1. 管理対象の **wpar1** ワークロード・パーティションを開始するには、以下のように入力します。

```
nim -o activate wpar1
```

2. 詳細出力を指定して開始するための **startwpar** コマンド・フラグを追加指定して、管理対象の **wpar1** ワークロード・パーティションを開始するには、以下のように入力します。

```
nim -o activate -a cmd_flags="-v" wpar1
```

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** というシステム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
```

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** 環境を初期化できます。

```
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・システム **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・システム **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 **/exports/resolv.conf** ファイルを作成します。

```
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
```

変更

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、 gateway0Z 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 gateway0Z" 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
```

chwpar

wpar1 ワークロード・パーティション用のリソース制御属性に rset rs/cpus23 を追加するには、以下のように入力します。

```
nim -o chwpar -a cmd_flags="-R rset=rs/cpu23" wpar1
```

create

1. ホスト名と仕様ファイル・リソース basic_wpar を指定して、wpar1 ワークロード・パーティションを作成するには、以下のように入力します。

```
nim -o create -a wpar_spec=basic_wpar wpar1
```

2. wpar-specification ファイル・リソース wpar1_spec を指定して、wpar1 ワークロード・パーティションを作成するには、次のように入力します。

```
nim -o create -a wpar_spec=wpar1_spec wpar1
```

3. savewpar バックアップ・イメージ・リソース wpar1_backup から wpar1 ワークロード・パーティションを作成するには、以下のように入力します。

```
nim -o create -a savewpar=wpar_backup wpar1
```

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
```

deactivate

1. 管理対象の `wpar1` ワークロード・パーティションを停止するには、以下のように入力します。

```
nim -o deactivate wpar1
```

- 管理対象の `wpar1` ワークロード・パーティションを強制的に停止するには、以下のように入力します。

```
nim -Fo deactivate wpar1
```

- 85 秒後に中断するための `stopwpar` コマンド・フラグを追加で指定して、管理対象の `wpar1` ワークロード・パーティションを停止するには、次のように入力します。

```
nim -o deactivate -a cmd_flags="-t 85" wpar1
```

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

- 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
```

- `image` という名前の 1 つの `lpp_source` を使用し、`/export/exec directory` の NIM マスターに `myspot` という名前の新しい SPOT リソースを作成するには、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
```

- 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
```

- 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
```

- 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
```

- `Disklsmacs1` という名前のマシン・グループを `diskls1`、`diskls2`、および `diskls3` 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=diskls1 ¥  
-a add_member=diskls2 -a add_member=diskls3 Disklsmacs1
```

- `DisklsRes1` 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 ¥  
DisklsRes1
```

- 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
```

9. **mksysb** リソース **mksysb2** を定義して、リソースの定義中にクライアント **client1** の **mksysb** イメージを作成する際、イメージがマスター上の **/resources/mksysb.image** にある場合、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
```

10. **mksysb** リソース **mksysb2** を定義して、リソースの定義中にクライアント **client1** の **mksysb** イメージを作成する際、イメージの作成に使用される **mksysb** フラグが **-em** で、イメージがマスター上の **/resources/mksysb.image** にある場合、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
```

11. マスターの **/resources/mksysb.image** 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
```

12. 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_speed1=16 redfish
```

13. 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。

14. **/export/nim/adapters** ディレクトリーを、マスター上の **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。

15. **savevg** リソース **savevg2** を定義するのに必要なスペースを表示して、リソースの定義中にクライアント **client1** の **savevg** イメージを作成する際、イメージがマスター上の **/export/nim/savevg** があり、バックアップする **volume_group** が **myvg** である場合、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です。 **savevg** またはリソース作成は行われません。

16. **savevg** リソース **savevg2** を定義して、リソースの定義中にクライアント **client1** の **savevg** イメージを作成する際、イメージがマスター上の **/export/nim/savevg** にあり、バックアップする **volume_group** が **myvg** である場合、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
```

17. **savevg** リソース **savevg2** を定義して、リソースの定義中にクライアント **client1** の **savevg** イメージを作成する際、イメージの作成に使用される **savevg** フラグが **-em** で、イメージがマスター上の **/export/nim/savevg** にある場合、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
```

18. ロケーション **/export/nim** にあるマスターに **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
```

19. **yogi** 管理元スタンドアロン・マシンが管理する **wpar1** ワークロード・パーティションを定義 (管理元システム上でホスト名とワークロード・パーティション名の両方として **wpar1** を指定) するには、以下のように入力します。

```
nim -o define -t wpar -a mgmt_profile1="yogi wpar1" -a if1="find_net wpar1 0" wpar1
```

20. **wpar1backup** という名の **savewpar** リソースを定義し、**sterling** サーバー上で **yogi** ワークロード・パーティションの **savewpar** イメージを作成するには、以下のように入力します。

```
nim -o define -t savewpar ¥
-a server=sterling -a location=/resources/wpar1.image ¥
-a source=wpar1 -a mk_image=yes wpar1backup
```

21. **wpar1backup** という名の **savewpar** リソースを定義し、**sterling** サーバー上で **yogi** ワークロード・パーティションの **savewpar** サーバーを作成するには (この場合、**exclude_files** リソースの **wparexclude** にあるファイル・パターンを除外し、各ファイルの除外と **image.data** ファイルの作成を行うために **savewpar** リソースにフラグを受け渡す)、以下のように入力します。

```
nim -o define -t savewpar ¥
-a server=sterling -a location=/resources/wpar1.image -a source=wpar1 ¥
-a exclude_files=wparexclude -a cmd_flags="-ei" mk_image=yes wpar1backup
```

22. **ios_mksysb** リソースを **ios_mksysb1** などのように定義し、リソース定義中に **vios** クライアントの **ios_mksysb** イメージを **vios1** として作成する (ここで、イメージはマスター上の **/export/nim/ios_mksysb** にある) には、次のように入力します。

```
nim -o define -t ios_mksysb -a server=master ¥
-a location=/export/nim/ios_mksysb -a source=vios1 ¥
-a mk_image=yes ios_mksysb1
```

destroy

1. 管理対象の **wpar1** ワークロード・パーティションを、その管理元システムから除去するには、以下のように入力します。

```
nim -o destroy wpar1
```

2. 管理対象の wpar1 ワークロード・パーティションを強制的に除去するには、以下のように入力します。

```
nim -Fo destroy wpar1
```

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、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=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
```

lswpar

1. 管理対象の wpar1 ワークロード・パーティション特性をリストするには、以下のように入力します。

```
nim -o lswpar wpar1
```

2. 管理対象の wpar1 ワークロード・パーティションのネットワーク特性をリストするには、以下のように入力します。

```
nim -o lswpar -a cmd_flags="-N" wpar1
```

3. global1 スタンドアロン・システムが管理するワークロード・パーティションの一般特性をリストするには、以下のように入力します。

```
nim -o lswpar -a cmd_flags="-G" global1
```

maint

1. 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
```

2. 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
```

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。

削除

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. 構成スクリプト 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
```

6. 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
```

7. **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
```

8. **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
```

syncwpar

1. 管理対象の `wpar1` ワークロード・パーティションのソフトウェアを、その管理元システムと同期するには、以下のように入力します。

```
nim -o syncwpar wpar1
```

2. `global1` スタンドアロン・システムが管理する全ワークロード・パーティションのソフトウェアを同期化するには、以下のように入力します。

```
nim -o syncwpar -a cmd_flags="-A" global1
```

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 7.1.0.0` 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 7.1.0.0 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
```

updateios

1. フィックスをインストールするか、`vioserver1` NIM オブジェクト名を指定して VIOS を最新の保守レベルに更新するには、次のように入力します。

```
nim -o updateios -a lpp_source=lpp_source1 -a preview=no vioserver1
```

更新は `lpp_source` ファイルおよび `lpp_source1` ファイルに保管されます。

注: **updateios** 操作では、インストール時にプレビューが実行されます。プレビュー・フラグが **no** に設定されていない限り、NIM から **updateios** 操作を実行すると、プレビューが実行されます。インストール時に、**updateios_flags=install** を指定して **updateios** 操作を使用する際、プレビューを実行する必要があります。プレビューを使用すると、VIOS 更新に進む前に、プレビュー・インストールが正確に実行されているかどうかを検査できます。

2. VIOS に対するフィックスを `vioserver1` NIM オブジェクト名を指定して拒否するには、次のように入力します。

```
nim -o updateios -a updateios_flags=reject vioserver1
```

3. VIOS に部分的にインストールされた更新を `vioserver1` NIM オブジェクト名を指定してクリーンアップするには、次のように入力します。

```
nim -o updateios -a updateios_flags=cleanup vioserver1
```

4. VIOS に対する更新を *vioserver1* NIM オブジェクト名を指定してコミットするには、次のように入力します。

```
nim -o updateios -a updateios_flags=-commit vioserver1
```

5. VIOS に対する **update1** などの特定の更新を *vioserver1* NIM オブジェクト名を指定して削除するには、次のように入力します。

```
nim -o updateios -a updateios_flags=-remove-a filesets="update1" vioserver1
```

6. VIOS に対する更新を *vioserver1* NIM オブジェクト名を指定し、*installp_bundle bundle1* (ここで、*bundle1* には、削除する更新が含まれています) を使用して削除するには、次のように入力します。

```
nim -o updateios -a updateios_flags=remove -a installp_bundle=bundle1 vioserver1
```

ファイル

項目	説明
<i>/etc/niminfo</i>	NIM が使用する変数が入っています。

関連資料:

144 ページの『*nimconfig* コマンド』

151 ページの『*niminit* コマンド』

関連情報:

lsnim コマンド

lssecattr コマンド

特権コマンド・データベース

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

項目	説明
-m <i>mksysb_resource</i>	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。 <i>mksysb_resource</i> 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。 <i>/export/nim/client.defs</i> 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 <i>/usr/samples/nim/client.defs</i> 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・オブジェクトをリブートします。デフォルトでは、クライアントはリブートされません。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ブートできます。
-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、クライアントをリブ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

項目	説明
<code>/etc/niminfo</code>	NIM が使用する変数が入っています。

関連資料:

- 109 ページの『`nim_master_setup` コマンド』
- 『`nim_master_recover` コマンド』
- 88 ページの『`nim` コマンド』
- 144 ページの『`nimconfig` コマンド』
- 147 ページの『`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。

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.sysmgt** 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.sysmgt.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.sysmgt** 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.sysmgt.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** ファイルについて詳しくは、「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

項目	説明
-D	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。
-f <i>directory</i>	インストールする bos.sysmgmt.nim.master ファイルセットが含まれているディレクトリ。
-i <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。
-n <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> 。
-N <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。"nim_net_name [hostname] [cable_type]" ここで、 <i>hostname</i> および <i>cable_type</i> はオプションです。
-p	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。
-r <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。
-R	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。
-S	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。
-s	各クライアント上の niminfo 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 bos.sysmgmt.nim.client 5.1.0.10 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 niminfo は更新されません。
-t <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 -N 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。"nim_net_name net_type net_addr net_snm default_route" 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。
-u	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。
-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_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。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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.sysmgmt.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** という名前の別の既存の 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** という名前がアドレスが **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

項目	説明
/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られます。

関連資料:

104 ページの『**nim_clients_setup** コマンド』

123 ページの『**nim_update_all** コマンド』

『**nim_master_setup** コマンド』

88 ページの『**nim** コマンド』

関連情報:

特権コマンド・データベース

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。
- 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

項目	説明
-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。

セキュリティー

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。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティー」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、`lssecattr` コマンドまたは `getcmdattr` サブコマンドの項を参照してください。

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

項目	説明
<code>/etc/niminfo</code>	NIM が使用する変数が入っています。
<code>/var/adm/ras/nim.setup</code>	コマンド実行結果のログ情報が入れられます。

関連資料:

- 104 ページの『`nim_clients_setup` コマンド』
- 106 ページの『`nim_master_recover` コマンド』
- 88 ページの『`nim` コマンド』
- 144 ページの『`nimconfig` コマンド』

関連情報:

lssecattr 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を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 ] ] [ -I 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 installp_bundle ] [ -f fix_bundle ] { [ -n ] [ -d ] } | -O [ -q ] [ -T ] [ -M manual_configuration_filenames ] }
```

説明

nim_move_up コマンドは、新しい POWER® Systems ハードウェア上で使用できる機能を既存の AIX 環境のユーザーが使用できるようにします。このコマンドは、既存の AIX システムを、POWER サーバーにある LPAR に移行するインターフェースを提供します。元のマシン上にある AIX のレベルは、より新しいハードウェア上の操作をサポートするレベルに高められます。元のシステムのハードウェア・リソースは、より新しいハードウェア上で厳密に複製されます。移行が終了するまでには、同じシステムが新規 LPAR 上で完全に稼働しています。

さらに、**nim_move_up** では、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) 上にクライアントを移行することにより、POWER サーバーの仮想入出力機能を使用できます。

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. POWER サーバーにある LPAR 上にバックアップをインストールする。これは NIM 環境では新規スタンドアロン・クライアントとして表されます。

新規ハードウェアのインストール前に、NIM マスター (**nim_move_up** コマンドが実行される場所) 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。このクライアントは移行の開始点であり、最終的には新規 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。
- 新規 POWER サーバー上の LPAR は元の NIM クライアントに対応していて NIM マスターによって制御されます。
- HMC は、SSH を使用して NIM マスターと通信することによって新規 POWER 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で行われます。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

項目	説明
-c <i>NIM_client</i>	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 (<i>mac_group</i> 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、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、POWER サーバー上の新規 LPAR 上に移行されるターゲット・マシンです。
-g <i>gateway</i>	これは、デフォルト・ゲートウェイの IP アドレスを指定します。クライアントは、POWER サーバーへの移行後に、この IP アドレスを使用して構成されます。
-h <i>control_host</i>	これは、POWER 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 <i>target_ip[ending_ip]</i>	これは IP アドレスを指定します。新たに移行されたクライアントは、POWER サーバーにインストールされた後で、このアドレスを使用して構成されます。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。
-I <i>img_src</i>	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。
-l <i>resource_dir</i>	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になります。 -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場合、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 <i>managed_sys</i>	これは、POWER サーバーに対応する管理対象システムの名前を、HMC で追跡されたとおりに指定します。
-s <i>subnet_mask</i>	これはサブネット・マスクを指定します。クライアントは、POWER サーバーへの移行後に、このマスクを使用して構成されます。

実行と制御のフラグ

項目	説明
-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

項目	説明
-b <i>installp_bundle</i>	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 <i>bosinst_data</i>	これは、新規クライアントを新規 LPAR にインストールするために nim_move_up で使用される既存の bosinst_data NIM リソースを指定します。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。
-C <i>script_resource</i>	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	これは、仮想入出力サーバー LPAR の指定時に POWER サーバー上で新規 LPAR を作成するときに仮想 SCSI アダプターではなく物理ストレージ・コントローラーの使用を強制します。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。

項目	説明
-e	これは、仮想入出力サーバー LPAR の指定時に POWER サーバー上で新規 LPAR を作成するときに共用イーサネット・アダプターではなく物理ネットワーク・アダプターの使用を強制します。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。
-M manual_configuration_filenames	phase4 で、関連するバックレベル AIX マシンに対してこれらの人手による構成ファイルを使用することを指定します。このフラグが有効になるのは、 nim_move_up コマンドの phase4 の中だけです。このフラグの詳細は、「拡張使用」セクションを参照してください。
-p loops	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。移行対象の LPAR の導出元となる同等の POWER サーバー・リソースを決定するときには、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。
-t seconds	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。
-T	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。
-u	nim_move_up が完全に nim_move_up 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 -r フラグと一緒に使用しなければなりません。
-U spot	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、 -L および -I オプションで指定した lpp_source NIM リソースから新しいスポットが作成されます。
-V vio_server	これは、 -m フラグで指示された POWER サーバー上にある仮想入出力サーバーの LPAR 名を指定します。

終了状況

項目	説明
0 nonzero	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

1. 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
```

2. **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
```

3. 保管済みのホスト名を新しい名前に変更して、出力を抑止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
```

4. 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
```

5. **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。

- 稼働中の 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) 以降
- 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 POWER サーバー構成をサポートするだけの十分なハードウェア・リソースを持つ POWER 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。

- POWER サーバーを制御する HMC。および電源のオン/オフを行ったり LPAR を作成したりするだけの十分な特権がその HMC にある。

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 コマンドはフェーズ方式を使用して既存のクライアントを新規 LPAR 上に移行します。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. **POWER** サーバー・リソース可用性データを収集し、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 がリブートされ、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 クライアントを新規 LPAR 上に正常に移行するために、各フェーズ (フェーズ 10 (オプション) を除く) を完全に首尾良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、新規移行済み LPAR の TCP/IP 構成情報、および POWER 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、* (アスタリスク) によっても示されています)。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・ライン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・ライン・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・ライン・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・ライン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・ラインの **-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・ライン・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。この編集可能なファイルは、コマンド・ライン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・ラインでは不可能なことは、**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・ラインの引数または **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 の次回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。POWER サーバーにおいて使用できる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。そのファイルに含まれるデータはすべて、POWER サーバー上に新たに導出された LPAR のハードウェア・プロファイルを決定するときに考慮されます。

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・ラインと 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、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。

人手による構成ファイルとは何か、およびこのファイルが必要な理由

nim_move_up コマンドの **phase4** で、このツールはバックレベル AIX マシンに基づいて各種リソース要件を計算し、ターゲット POWER サーバー上に LPAR を適切に作成します。お客様が、以下の状態のいずれかである場合、人手による構成ファイルに、必要な変更を事前定義されたフォーマットを使用して指定し、**nim_move_up** コマンドを実行することができます。

- **nim_move_up** コマンドで決定されたメモリよりも多くのメモリが必要である。
- ボリューム・グループ用に使用したい仮想入出力サーバー上に作成された仮想 SCSI アダプター (**vhost#**) がある。
- **nim_move_up** ツールで生成された仮想ローカル・エリア・ネットワーク (VLAN) ID とは異なる ID を使用したい。

nim_move_up コマンドの正常終了後、人手による構成がターゲット LPAR に適用されます。

人手による構成ファイルの作成方法

注: **nim_move_up** コマンドを開始する前に、人手による構成ファイルを作成する必要があります。マイグレーション対象のクライアントごとにこれらのファイルを作成し、**nim_move_up** コマンドの **-M** フラグの引数としてこれらのファイルを指定することにより、このコマンドで人手による構成を使用することができます。このファイル名の形式は、**path/manual_cfginfo_client_host_name** にする必要があります。ここで、**path** 値は、人手による構成ファイルがあるディレクトリーであり、**client_host_name** 値は、マイグレーション対象のクライアント・マシンのホスト名です。

nim_move_up コマンドは、POWER Systems 環境に移行されるクライアントごとにデフォルトでハードウェア構成関連の計算をすべてを行います。このファイルを使用すると、お客様が選択するとおりにターゲット・マシンの構成の変更またはチューニングが行えます。

メモリー量、ボリューム・グループのサイズ、および使用対象の仮想入出力サーバー・リソースを変更できます。例えば、ターゲット LPAR 用に作成されたボリューム・グループとして使用する VSCSI サーバー・アダプターを変更できます。また、ターゲット LPAR 用に作成されたイーサネット・アダプターとして使用する VLAN ID も変更できます。

以下の例が、人手による構成ファイルです。

```
# manual_cfgfile_dennis file
# MEMORY = min_MB desired_MB max_MB
MEMORY = 256 512 1024
# VIO_VG_INFO = vname_src, size_in_MB, vhost_to_use
#   Where vname_src is the VG name in source machine, and
#   vhost_to_use is the virtual adapter to be used for
#   the VG specified in the VIO Server.
VIO_VG_INFO = rootvg,15344,vhost4
# VIO_VLAN_INFO = vlan_id, lpar_name, slot_number
VIO_VLAN_INFO = 1,VIO-server,2
```

このファイルには空白行が入ってもかまいません。行の先頭に # を 1 つ指定して、このファイルにコメントを追加できます。

min_MB、*desired_MB*、および *max_MB* の各値のすべてはメガバイト (MB) 単位にする必要があります。これらの数値の間に入れるスペース数に関する制約はありません。

min_MB

AIX の実行に必要なとなる最小メモリー。

desired_MB

論理パーティションが活動化された時点でそのパーティションに保有させたいメモリー量。

max_MB

動的論理パーティション操作がそのパーティション上で行われる時の最大メモリー。

VIO_VG_INFO フィールドの各値はコンマで区切る必要があります。*vname_src* 値は、人手によるデータを指定する対象ソース・マシンのボリューム・グループです。*size_in_MB* 値はターゲット・マシン上のボリューム・グループのサイズです。*vhost_to_use* 値は、ターゲット POWER サーバー上でこのボリューム・グループに使用される *vhost** (仮想 SCSI サーバー・アダプター) です。

同様に、VIO_VLAN_INFO フィールドの各値はコンマで区切る必要があります。*vlan_id* 値は、ターゲット LPAR のイーサネット・アダプターに対して、**nim_move_up** コマンドが使用する値の代わりに使用されます。*lpar_name* 値は、共用イーサネット・アダプター (SEA) を装備した仮想入出力サーバーの LPAR 名であり、*slot_number* 値は、その仮想入出力サーバー上のこの SEA のスロット番号です。

これらの値は、必ずしもすべて指定する必要はありません。**nim_move_up** コマンドは、指定された値を手書きファイルから受け取り、クライアント構成に基づいてその残りの値を生成します。

ファイル

項目
/usr/sbin/nim_move_up

説明
nim_move_up コマンドが入っています。

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

項目	説明
-d <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 /dev/cd0 です。
-l <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、 basic_res_grp から取得されます。
-s <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、 basic_res_grp から取得されます。
-B	basic_res_grp 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。
-u	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。
-v	コマンド実行時の詳細デバッグ出力を有効にします。セキュリティー

位置

/usr/sbin/nim_update_all

終了状況

成功するとゼロ (0) を返します。

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

1. 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
```

2. 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
```

3. 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

項目	説明
/etc/niminfo	NIM が使用する変数が入っています。
/var/adm/ras/nim.update	コマンド実行結果のログ情報が入れられます。

関連資料:

- 109 ページの『**nim_master_setup** コマンド』
- 104 ページの『**nim_clients_setup** コマンド』
- 106 ページの『**nim_master_recover** コマンド』
- 88 ページの『**nim** 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 ガイドの「NIM マスターの構成と基本インストール・リソースの作成」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・エントリーがキーワード **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` であるスタンザは無視されます。

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.sysmgt/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` コマンドに使用されます。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.sysmgt/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` コマンドに使用されます。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

項目	説明
-a	次の <code>attribute=value</code> 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 <code>info</code> 属性を使用する必要があります。 <code>AttributeList</code> は、属性をコマンド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： <code>interface_type</code> 、 <code>location</code> 、 <code>interface_name</code> 、 <code>cable_type</code> 、 <code>media_speed</code> 、 <code>netaddr</code> 、 <code>subnet_mask</code> 、 <code>interface_attributes</code> 、 <code>secondary_hostname</code> 、 <code>machine_type</code> 、 <code>adapter_attributes</code> 、 <code>multiple_physloc</code> 、 <code>bos_preconfig</code> 、 <code>cust_preconfig</code> 、 <code>route</code> 。値が使用されないことを指定するには、小文字の <code>n/a</code> 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 <code>Client.adapter</code> ファイルが <code>adapter_def</code> ロケーションに作成されます。 <code>nimadapters</code> 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	<code>SecondaryAdapterFileName</code> 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 <code>info</code> 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。 <ul style="list-style-type: none"> • 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 <code>-p</code> 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。 <code>client</code> 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、 <code>adapter_def</code> 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

項目	説明
<code>adapter_def</code>	<code>nimadapters</code> 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 <code>adapter_def</code> NIM リソースを指定します。 <code>nimadapters</code> コマンドで <code>adapter_def</code> を使用するためには、その前に <code>nim -o define</code> 操作を使用することによって <code>adapter_def</code> 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

アクセス制御: `nimadapters` コマンドを実行するには、`root` 権限が必要です。

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、`lssecattr` コマンドまたは `getcmdattr` サブコマンドの項を参照してください。

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

項目	説明
<code>/usr/sbin/nimadapters</code>	<code>nimadapters</code> コマンドが入っています。

関連資料:

144 ページの『`nimconfig` コマンド』

147 ページの『`nimdef` コマンド』

関連情報:

`lsnim` コマンド

`tunehange` コマンド

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アクション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 (複数の場合もある) に作成し、同時にこれを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。
- `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 mkysyb 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 mkysyb リソースを作成する。
- NIM mkysyb 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 ]
```

mkysyb からクライアントへのマイグレーション:

```
nimadm -T NIMmkysyb -c NIMClient -s SPOT -l lpp_source -d TargetDisks -j VGname -Y [ -a PreMigrationScript ] [ -b installp_bundle ] [ -z PostMigrationScript ] [ -i ImageData ] [ -m NFSMountOptions ] [ -o bosinst_data ] [ -P Phase ] [ -F ] [ -D ] [ -E ] [ -V ] [ -B | -r ]
```

mkysyb から mkysyb へのマイグレーション:

```
nimadm -T NIMmkysyb -O mkysybfile -s SPOT -l lpp_source -j VGname -Y [ -N NIMmkysyb ] [ -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 mkysyb へのマイグレーション:

```
nimadm -c nim_client -O mkysybfile -s SPOT -l lpp_source -j VGname -Y [ -N NIMmkysyb ] [ -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 コマンドは、ネットワーク管理者が **rootvg** のコピーを空きディスク (複数の場合もある) に作成し、同時にこれを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ための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、従来のマイグレーションに比べて次のような利点があります。

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ので、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。新しい (マイグレーションされた) レベルの AIX で問題が起こった場合、元のディスクからブートすると、システムをマイグレーション以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、オプションの 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) を使用して行われます。

本書は、**nimadm** コマンドに関する情報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。

- ガイドの「インストールおよび移行」
- AIX バージョン 4.3 から 5L への移行ガイド (IBM® Redbooks 資料の 1 つ)

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. 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。)
2. 選択した **lpp_source** NIM リソースおよび選択した SPOT NIM リソースは、マイグレーション先の AIX レベルと一致する必要があります。
3. 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。
4. 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。
5. 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。
6. 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。
7. 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。
9. 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。
10. 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。
11. **nimadm** コマンドを実行してクライアント・システムの **rootvg** を複製する前に、DB2 や LDAP など、すべてのアプリケーション・サーバーとデータベース・サーバーを停止する必要があります。そうしないと、**nimadm** コマンド操作の完了後に、アプリケーション・サーバーやデータベース・サーバーが正常に始動しません。

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** のトラステッド・コンピューティング・ベース (TCB) がオンになっている場合、これを (永久に) 使用不可にするか、ディスク・キャッシング・オプション (**-j**) を使用するか、ま

たは従来のマイグレーションを実行する必要があります。TCB はファイル・メタデータにアクセスする必要がありますが、ネットワーク・ファイルシステム (NFS) からはアクセス不能であるため、この制限が存在します。

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** ボリューム・グループ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** コマンドのクローン作成に関連した CPU 使用の増加により、クライアントに多少のパフォーマンス低下が生じる可能性があります。
4. **nimadm** のパフォーマンスを最適化するために NFS の調整が必要な場合があります。

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。**nimadm** フェーズは順次に行われる必要があります。**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** ボリューム・グループ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、スクリプト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・マージ方式が実行され、いくつかの各種処理が行われず。
6. システムのファイルセットは、**installp** を使用してマイグレーションされます。必要な RPM イメージはすべて、このフェーズ中に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、スクリプト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論理ボリューム (**hd5**) に書き込まれるクライアントのブート・イメージが作成されます。
9. フェーズ 3 でマスターに行われたマウントを取り外します。
10. フェーズ 2 で作成されたクライアント・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** をスリープにします。**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、クライアントはリブートされます。

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行を行うものであり、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

項目	説明
-a <i>PreMigrationScript</i>	pre-migration NIM スクリプト・リソースを指定します。
-b <i>installp_bundle</i>	installp_bundle NIM リソースを指定します。
-B	nimadm マイグレーションの後に bootlist を実行しないことを指定します。これを設定した場合は、 -r フラグは使用できません。
-c <i>ClientDisks</i>	この nimadm 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 nimadm 操作に必要です。
-C	nimadm クリーンアップを実行します。
-d <i>TargetDisks</i>	altinst_rootvg (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。
-D	nimadm コマンドをデバッグ・モードに設定します。この機能は nimadm 関連の問題をデバッグする時のみ使用し、デフォルトにすべきではありません。
-e <i>exclude_files</i>	exclude_files NIM リソースを指定します。このリソースは alt_disk_install コマンドでフェーズ 1 中に使用されます。
-E	重大なマイグレーション・エラーが起こった場合、 nimadm デバッガーに入ります。
-F	クライアントを強制的にアンロックします。通常、 nimadm 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 nimadm 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 nimadm 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。
-i <i>image_data</i>	image_data NIM リソースを指定します。このリソースは alt_disk_install 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。
-j <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。
-l <i>lpp_source</i>	この nimadm 操作で使用される lpp_source 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。
-m <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、 mount コマンドに渡す引数を指定します。このフラグは、 nimadm 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。
-M	NIM マスター、SPOT、 lpp_source 、およびオプション・デバイス上の alt_disk_install ソフトウェア (bos.alt_disk_install) 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、 nimadm コマンドは、 lpp_source 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。
-N <i>NIMmksysb</i>	作成する固有の新規 NIM mksysb リソースを指定します。 -N フラグを指定した場合は、 -O フラグも指定する必要があります。
-o <i>bosinst_data</i>	bosinst_data NIM リソースを指定します。
-O <i>mksysbfile</i>	マイグレーション済み mksysb のファイル・パス名を指定します。 -O フラグを指定した場合は、 -j フラグ、および -c か -T のどちらかのフラグを指定する必要があります。
-P <i>Phase</i>	nimadm 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。
-r	nimadm マイグレーションが完了した後、クライアントがリブートすべきことを指定します。
-s <i>SPOT</i>	この nimadm 操作に使用される SPOT NIM リソースを指定します。このフラグはすべての nimadm 操作に必要です。

項目	説明
-S	nimadm の「スリープ」機能を実行します。この機能は、 nimadm 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。
-T <i>NIMmksysb</i>	マイグレーションする既存の NIM <i>mksysb</i> リソースを指定します。-T フラグを指定した場合は、-j フラグ、および -O か -c のどちらかのフラグを指定する必要があります。
-V	詳細出力をオンにします。
-W	nimadm の「ウェイクアップ」機能を実行します。
-Y	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。
-z <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

アクセス制御: **nimadm** コマンドを実行するには、root 権限が必要です。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

1. 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
```

2. 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
```

3. 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
```

4. 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
```

5. 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
```

6. 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

項目	説明
/usr/sbin/nimadm	nimadm コマンドが入っています。

関連資料:

88 ページの『nim コマンド』

関連情報:

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

項目	説明
-a <i>Attribute=Value</i>	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 証明書を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 セキュリティーを使用します。
-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 <i>Lsnim parameters</i>	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 <i>Operation</i>	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。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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
```

- 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
```

- 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
```

- 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
```

- 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
```

- 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
```

- 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"
```

- 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** があり、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
```

- 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
```

- NIM スタンドアロンから、**lpp_source** **lpp_source1** にある APAR IX12345 のライブ更新を実行するには、次のように入力します。

```
nimclient -o allocate -a lpp_source=lpp_source1  
nimclient -o cust -a live_update=yes -a filesets=IX12345
```

- 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
```

- 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。

13. 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
```

14. 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
```

15. 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。

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_source** client_images を使用して NIM **installp_bundle** client_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

項目	説明
/etc/niminfo	NIM が使用する変数が入っています。

関連資料:

88 ページの『nim コマンド』

144 ページの『nimconfig コマンド』

151 ページの『nimitools コマンド』

関連情報:

lsnim コマンド

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

- | 項目 | 説明 |
|-----------|---|
| -a | 次の <code>attribute=value</code> 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ンネル・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ンネル・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

rspcsmp
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・ラインでポート番号を指定しない場合は、NIM の <code>/etc/services</code> ファイルのポート番号が使用されます。 <code>/etc/services</code> ファイルに NIM ポート <code>nim</code> および <code>nimreg</code> 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 <code>/etc/niminfo</code> ファイルを再作成します。このマシンで bos.sysmgt.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、 <code>/etc/niminfo</code> ファイルを間違っ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。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の

『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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せん。

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

項目	説明
/etc/niminfo	NIM が使用する変数が入っています。

関連資料:

88 ページの『nim コマンド』

140 ページの『nimclient コマンド』

151 ページの『niminit コマンド』

関連情報:

lsnim コマンド

.info コマンド

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

項目	説明
cable_type gateway	マシンのケーブル・タイプを指定します。 network_type が ent である場合に必須です。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 0 (ゼロ) を指定します。
machine_type network_type	マシンのタイプを指定します。 standalone 、 diskless 、または dataless です。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。 ent または tok です。
ring_speed subnet_mask	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。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

項目	説明
nim_name	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、競合が起こります。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。
platform	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 rs6k です。
net_adptr_name	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 (tok0 、 ent0 、など)。
netboot_kernel=NetbootKernelType	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。 netboot_kernel 値は up または mp です。
ipl_rom_emulation	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 (/dev/fd0 、 /dev/rmt0 など)。
primary_interface	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。
master_gateway	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。
machine_group comments	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・タイプのみ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

項目	説明
-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 定義スタンプ。

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プおよび障害の原因。

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プ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。

項目	説明
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: **nimdef** コマンドを実行するには、root 権限が必要です。

イベントの監査: N/A

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

1. 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
```

2. 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
```

3. **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

項目	説明
<code>/usr/sbin/nimdef</code>	nimdef デモン/コマンドが入っています。

関連資料:

88 ページの『nim コマンド』

140 ページの『nimclient コマンド』

144 ページの『nimconfig コマンド』

関連情報:

NIM の構成

lssecattr 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 iprom_emu=Device ] [ -a platform=PlatformType ] [ -a netboot_kernel=NetbootKernelType ] [-a adpt_add=AdapterAddress] [ -a is_alternate= yes | no ] [ -a connect=value ] [ -a vlan_tag=value ] [-a vlan_pri=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

項目	説明	属性の説明
-a	nimit 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 -a フラグが付いています。	
name=Name		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。
pif_name=Pif		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。
master=Hostname		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。
master_port=PortNumber		NIM コミュニケーションに使用する nimesis デーモンのポート番号を指定します。
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。受け入れ可能な値は、 bnc 、 dix 、および N/A です。
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。受け入れ可能な値は、 4 および 16 です。
iplrom_emu=Device		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。
platform=PlatformType		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 chrp 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。
	chrp	PowerPC Common Hardware Reference Platform (CHRP) アーキテクチャー・ベースのマシン
	rs6k	AIX 5.1 およびそれ以前用のマイクロチャンネル・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ンネル・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・マシン
	rspcsmp	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
adpt_add=AdapterAddress		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。
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:
		1. コマンド・ライン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。
		2. master_port および registration_port 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 lsnim -l master を実行します。

項目	説明	属性の説明
<code>netboot_kernel=NetbootKernelType</code>		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。 <code>netboot_kernel</code> の値は次のとおりです。 <code>up</code> ユニプロセッサ・マシンのカーネル。 <code>mp</code> マルチプロセッサ・マシンのカーネル。 デフォルト値は <code>up</code> です。
<code>is_alternate=[yes no]</code>		このマシンを <code>alternate_master</code> として構成する必要がある場合は、「yes」に設定します。
<code>connect=value</code>		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、 <code>shell</code> (<code>rsh</code> の場合) および <code>nimsh</code> です。デフォルト設定は <code>connect=shell</code> です。この属性はオプションです。 <code>is_alternate</code> 属性が <code>yes</code> に設定される場合、 <code>nimsh</code> はデフォルトの設定であり、唯一の有効な値です。 <code>is_alternate</code> 属性の使用はオプションです。
<code>vlan_tag=value</code>		VLAN タグ付けに使用する仮想論理エリア・ネットワーク (VLAN) の ID を指定します。ID はイーサネット・フレームが属する必要がある VLAN を識別するために使用されます。この ID により、ネットワーク管理者はネットワークをサブネットに割り当てるのではなく、論理的にクライアントの通信を編成することができます。VLAN タグ付けの値は NIM によって、クライアントのネットワーク・ブートを実行するために使用されます。VLAN タグ通信の構成は、値を使用する前に NIM の外部で処理する必要があります。有効な値は 0 から 4094 です。
<code>vlan_pri=value</code>		VLAN タグ付けに使用する仮想論理エリア・ネットワーク (VLAN) の優先順位を指定します。優先順位の値は、VLAN タグとともに、イーサネット・フレームが属する必要がある VLAN を識別するために使用されます。この優先順位により、ネットワーク管理者はネットワークをサブネットに割り当てるのではなく、論理的にクライアントの通信を編成することができます。VLAN タグ付けの値は NIM によって、クライアントのネットワーク・ブートを実行するために使用されます。 VLAN タグ通信の構成は、値を使用する前に NIM の外部で処理する必要があります。有効な値は 0 から 7 です。

セキュリティ

アクセス制御: `niminit` コマンドを実行するには、`root` 権限が必要です。

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、`lssecattr` コマンドまたは `getcmdattr` サブコマンドの項を参照してください。

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。

```
niminit -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。

```
niminit -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=rspcsmp
```

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。

```
nimit -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。

```
nimit -a is_alternate=yes -a name=octopus -a pif_name=en0 ¥  
-a cable_type=bnc -a master=dolphin
```

ファイル

項目	説明
<code>/etc/niminfo</code>	NIM が使用する変数が入っています。

関連資料:

88 ページの『`nim` コマンド』

140 ページの『`nimclient` コマンド』

144 ページの『`nimconfig` コマンド』

関連情報:

`lsnim` コマンド

`.info` コマンド

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=yes|no ]
```

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=yes|no ]
```

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=yes|no ] [ -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

項目	説明
-a <i>attribute=value</i>	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 <i>operation</i>	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。

項目
-o operation (続き)

説明

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せん。

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 **conglomeratebase_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。

項目	説明
-o operation (続き)	<p>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。</p> <p>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ンドで区切られたリスト。</p> <p>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。</p> <p>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。</p> <p>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。</p>
	<p>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。</p> <p>targets (必須) インストール・インベントリーを集めるための、コマンドで区切られた NIM オブジェクトのリスト。</p> <p>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。</p> <p>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。</p>

終了状況

項目	説明
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: **niminv** コマンドを実行するには、**root** 権限が必要です。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: **rbacqry** コマンドは、**root** ユーザーに実行 (x) アクセス権限を与えます。**rbacqry** コマンドは、特権操作の実行に使用される特権コマンドです。コマンドを正常に実行するには、権限をもつロールをアクティブにする必要があります。

例

1. 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。

```
niminv -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 **lslpp -L** の出力に似ています。

2. 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。

```
niminv -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 **lslpp -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・インベントリーを、**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。

```
niminv -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・インベントリーのみです。

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。

```
niminv -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・インベントリーが含まれます。

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。

```
niminv -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 コマンド

lslpp コマンド

lsmcode コマンド

「Software Service Management (ソフトウェア・サービス管理)」メニューの使用 (SUMA を含む)

suma コマンド

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

項目	説明
<code>-c client_hostname</code>	<code>geninstall</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>-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>	<code>geninstall</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。
<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

項目	説明
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

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` ディレクトリー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、`dhcpd.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、`-l` 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

項目	説明
<code>-C</code>	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。
<code>-d directory</code>	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 <code>/mnt/cdrom</code> です。
<code>-D</code>	コマンドをデバッグ・モードで実行します。
<code>-e</code>	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。
<code>-l</code>	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。
<code>-L label</code>	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。
<code>-m method</code>	<code>nimol_install</code> コマンドに対して <code>-n</code> 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。
<code>-r</code>	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。
<code>-s hostname</code>	NIMOL サーバーに使用するホスト名。デフォルトでは、 <code>hostname</code> コマンドを実行することによりホスト名を決定します。
<code>-S</code>	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時、状況はログに記録されません。

項目	説明
-t <i>directory</i>	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

項目	説明
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

項目	説明
/etc/nimol.conf	コマンドの構成情報を保管します。

nimol_install コマンド

目的

構成済み NIMOL サーバーをセットアップして、特定のクライアント・マシンに AIX をインストールします。

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 をマシンにネットワーク・インストールするために、構成済みの 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い場合、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い場合、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、**/tftpboot** 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 **/tftpboot** ディレクトリー内に作成されます。
11. **-l** 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稼働中の場

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、AIX クライアントが 192.168.1.30 である場合、**ent0** をブートオフするには、次のコマンドを実行します。

```
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

項目	説明
-c <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。
-D	コマンドをデバッグ・モードで実行します。
-g <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。
-l	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。
-L <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 default です。
-m <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、00:60:08:3F:E8:DF)。
-n	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。
-p <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。
-r	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 をインストールできなくなります。このフラグにはクライアント・ホスト名が必要です。
-s <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

項目	説明
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である場合 (**-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

項目	説明
<code>/etc/nimol.conf</code>	コマンドの構成情報を保管します。

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 `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 `lslpp` コマンドを実行します。 `lslpp` 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

項目	説明
-c <i>client_hostname</i>	lspp 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。
-D	コマンドをデバッグ・モードで実行します。
-f <i>lspp_flags</i>	lspp コマンドに渡す、 lspp コマンド・フラグを指定します。
-m <i>remote_access_method</i>	lspp 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 /usr/bin/rsh です。もう 1 つのオプションは、 /usr/bin/ssh です。

終了状況

項目	説明
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

項目	説明
/etc/nimol.conf	コマンドの構成情報を保管します。

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

項目	説明
-c <i>client_hostname</i>	geninstall 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。
-D	コマンドをデバッグ・モードで実行します。
-f <i>geninstall_flags</i>	geninstall 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、 acgX です。
-L <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。
-m <i>remote_access_method</i>	geninstall 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。
-p <i>package_list</i>	geninstall 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

項目	説明
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

- 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
```

- 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"
```

- `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

項目	説明
<code>/etc/nimol.conf</code>	コマンドの構成情報を保管します。

nimquery コマンド

目的

システム情報についてネットワーク・インストール管理 (NIM) 環境のシステムに照会し、この環境でクライアント・オブジェクトを作成します。

構文

```
nimquery{-a host=hostname [-a name=client obj[-d] [-a hmc=obj name[-d] [-a cec=obj name [-a bcmm=obj name [-a ivm=obj name] [-p] [-q] [-v]
```

説明

nimquery コマンドは、**-a** ホスト・パラメーターを使用している場合、システム情報についてマシンに照会します。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 サービス・ハンドラー (nimsh) を使用するシステムから提供されます。

nimquery コマンドは、ハードウェア管理コンソール (HMC)、中央電子処理装置 (CEC)、Integrated Virtualization Manager (IVM)、または Blade Center 管理モジュール (BCMM) 指している場合に、論理区画 (LPAR)、CEC、およびブレードの情報を照会するために使用することもできます。これを行うには、**openssh.base.client** を NIM マスターにインストールする必要があります。

フラグ

項目	説明
-a	以下のパラメーター属性=値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 (-a ホストを使用する場合は name 属性が必要です)。
-p	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

項目	説明
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。
hmc=objname	照会する HMC システムのオブジェクト名を指定します。この属性は必要です。
cec=objname	照会する CEC システムのオブジェクト名を指定します。この属性は必要です。
ivm=objname	照会する IVM システムのオブジェクト名を指定します。この属性は必要です。
bcmm=objname	照会する BCMM システムのオブジェクト名を指定します。この属性は必要です。

終了状況

0 正常時にゼロを戻します。

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

- 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
```
- システム情報についてマシン `buckey` に照会し、詳細な出力情報を提供するには、次のように入力します。

```
nimquery -a host=buckey -p
```
- 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
```
- ブレードのシステム情報について管理モジュール `bcmm2` に照会するには、次のように入力します。

```
nimquery -a bcmm=bcmm2
```
- HMC `hmc1` が管理する CEC オブジェクトを定義するには、次のように入力します。

```
nimquery -a hmc=hmc1 -d
```
- システム情報について `cec1 buckey` に接続された LPAR に照会するには、次のように入力します。

```
nimquery -a cec=cec1
```

ファイル

項目	説明
<code>/usr/sbin/nimquery</code>	nimquery コマンドの場所。

関連資料:

- 88 ページの『nim コマンド』
- 144 ページの『nimconfig コマンド』
- 147 ページの『nimdef 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・ラインにサーバー情報 (**-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 **mksecdap** 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。2³¹-1 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

項目	説明
-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、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、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せん。
-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、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、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 のどちらかを指定できます。RFC2307AIX は、拡張 AIX スキーマ・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。

項目	説明
-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

- 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
```

- 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
```

- /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
```

- 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
```

- 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、次のコマンドを実行します。

```
ldapmodify -f /tmp/ch_user2.ldif
```

ファイル

項目	説明
/usr/sbin/nistoldif	nistoldif コマンドが入っています。

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・ライン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・ラインではファイルを 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。

項目	説明
-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。 a すべての行に番号を付けます。 t 空白行や、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) n どの行にも番号を付けません。 p <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。
-d <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 ¥: (円記号、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。 -d 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。円記号を区切り文字として使用する場合は円記号を 2 つ (¥¥) 入力します。
-f <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 フラグと同じです。 <i>Type</i> 変数のデフォルト値は n です (行番号を付けない)。
-h <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 フラグと同じです。 <i>Type</i> 変数のデフォルト値は n です (行番号を付けない)。
-i <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。
-l <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、 -l3 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、 -ha 、 -ba 、または -fa 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。
-n <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。 ln 左に揃い、先行ゼロが抑制されます。 rn 右に揃い、先行ゼロが抑制されます (デフォルト)。 rz 右に揃い、先行ゼロはそのままになります。
-p	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。
-s <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。
-v <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。
-w <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。

項目	説明
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。

```
n1 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す。

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。

```
n1 -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。

- 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

項目	説明
<code>/usr/bin/nl</code>	nl コマンドを含みます。

関連資料:

446 ページの『**pr** コマンド』

関連情報:

ファイル・コマンド

入出力ダイレクト

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。AIX プラットフォームの場合、**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

項目	説明
-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 <i>group_name</i>	状況を入手するサブシステムのグループを指定します。 <i>group_name</i> 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 <i>host</i>	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 (<i>/etc/inittab</i> を参照) を -r フラグを使用して開始し、 <i>/etc/hosts.equiv</i> ファイルまたは <i>.rhosts</i> 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 <i>subsystem_pid</i>	状況を入手する <i>subsystem_pid</i> 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 <i>subsystem_name</i>	状況を入手するサブシステムを指定します。 <i>subsystem_name</i> 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。 <i>subsystem_name</i> 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

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

Group Name Number of Number of local
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

`/opt/rsct/bin/nlssrc`

`nlssrc` コマンドが入っています。

ファイル

`/tmp/ctsupt`

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/ctsupt/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/ctsupt/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。

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)。

項目	説明
A	グローバル絶対シンボル
a	ローカル絶対シンボル
B	グローバル bss シンボル
b	ローカル bss シンボル
D	グローバル・データ・シンボル
d	ローカル・データ・シンボル
f	ソース・ファイル名シンボル
L	グローバル・スレッド・ローカル・シンボル (TLS)
l	静的スレッド・ローカル・シンボル (TLS)
T	グローバル・テキスト・シンボル
t	ローカル・テキスト・シンボル
U	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

項目	説明
-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、使用され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。
-l	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	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せん。
-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。
-r	逆順にソートします。
-t <i>Format</i>	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ません。
-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 <i>mode</i>	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。

項目	説明
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

項目	説明
/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

a.out コマンド

ar コマンド

commands コマンド

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 < seconds > ] [ -c < count > ] [ -b ] [ -B ] [ -g < filename > ] [ -k disklist ] [ -C < process1:process2...:processN > ] [ -i ]
```


記録モード:

```
nmon [ -f | -F filename | -x | -X | -z ] [ -r < runname > ] [ -t | -T | -Y ] [ -s seconds ] [ -c number ] [ -w number ] [ -l dpl ] [ -d ] [ -g filename ] [ -k disklist ] [ -C <process1:process2:...:processN > ] [ -G ] [ -K ] [ -o outputpath ] [ -D ] [ -E ] [ -J ] [ -V ] [ -P ] [ -M ] [ -N ] [ -W ] [ -S ] [ -^ ] [ -O ] [ -L ] [ -I percent ] [ -A ] [ -m < dir > ] [ -Z priority ] [ -i ] [ -y options ]
```

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。このコマンドは対話モードまたは記録モードのいずれかで実行できます。 **-F**、**-f**、**-X**、**-x**、および **-Z** フラグのいずれを指定した場合も、**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。

- アダプター入出力統計(**a** キーを使用)
- AIO プロセス・ビュー (**A** キーを使用)
- 詳細ページ統計 (**M** 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)
- 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)
- ESS vpath 統計ビュー (**e** 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)
- 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)
- メモリーとページング統計 (**m** 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)
- NFS パネル (**N** キーを使用)
- ページ・スペース (**P** キーを使用)
- プロセス・ビュー (**t** と **u** 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)
- 共用イーサネット・アダプター統計 (**O** 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)
- システム・リソース・ビュー (**r** キーを使用)
- スレッド・レベルの統計 (**i** キーを使用)
- 詳細検査 OK/警告/危険のビュー (**v** キーを使用)

- ボリューム・グループ統計 (**V** キーを使用)
- WLM ビュー (**W** キーを使用)

記録モードでは、このコマンドは **nmon** ファイルを生成します。これらのファイルは、それをオープンすることにより、あるいは **nmon** アナライザーなどのポストプロセッシング・ツールを使用して、直接表示できます。**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 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・ラインから **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 (オンラインで、およびファイルに対して) リストするプロセスを制限するには、以下のオプションを利用することができます。

- 環境変数内のプログラム名を **NMONCMD0** から **NMONCMD63** までで設定する。
- `cmd:cmd:cmd` パラメーター指定で **-C** フラグを使用する。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* 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。オンデマンド記録機能の使用がきっかけとなって記録が行われた場合は、この機能は正しくありません。

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。

workload partitions (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・チャネル・アダプター
- VIOS 共用イーサネット・アダプター

注: システムに適用される動的構成変更は、現行の **nmon** の記録には反映されません。新しい構成変更を有効にするには、**nmon** ツールを再始動する必要があります。

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。

項目	説明
-s < <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。
-c < <i>count</i> >	画面をリフレッシュする必要がある回数。
-g < <i>filename</i> >	<i>filename</i> パラメーターを使用して指定可能な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	モノクロ・モードでビューを表示します。
-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。
-h	ヘルプ情報を表示します。
-k < <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。
-i	トップ・スレッド・レベルの CPU 使用率を報告します。

記録モードでのフラグ

項目	説明
-A	ビューに非同期入出力セクションを組み込みます。
-c	このコマンドにより採取されるスナップショット数を指定します。デフォルト値は 10000000 です。
-d	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。
-D	ディスク構成セクションをスキップします。
-E	ESS 構成セクションをスキップします。
-f	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、このコマンドは各スナップショット間で 300 秒の間隔をおいてシステム・データの 288 個のスナップショットを採取します。出力ファイル名は、 <i>hostname_YYMMDD_HHMM.nmon</i> の形式になっています。
-F	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。
-g	<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。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。
-G	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 つのシステムにある多くの LPAR から nmon 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。
-i	スレッド・レベルの統計を報告します。
-I	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。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、コマンドは TOP プロセス統計を保存しません。
-J	JFS セクションをスキップします。
-k	記録対象のディスク・リストを指定します。
-K	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。 -K 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。メモリー・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。
-l	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。
-L	EMC ディスクの場合、64 の値を指定してください。
-L	ラージ・ページ分析セクションを組み込みます。
-m	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。

項目	説明
-M	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。
-N	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、 -NN を指定します。
-o	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。
-O	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。
-P	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。
-r	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。
-s	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。
-S	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。
-t	トップ・プロセスを出力に組み込みます。 -t 、 -T 、または -Y フラグを同時に指定できません。
-T	トップ・プロセスを出力に組み込み、コマンド・ライン引数を UARG セクション保存します。 -t 、 -T 、または -Y フラグを同時に指定できません。
-V	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。
-w	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、 .csv 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。
-W	記録ファイルに WLM セクションを組み込みます。
-x	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、 -ft -s 900 -c 96 と同じです。
-X	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、 -ft -s 30 -c 120 と同じです。
-y options	nmon 記録セクションを制御します。 options パラメーターの値は、コマンドで区切る必要があります。

options パラメーターの有効な値は次のとおりです。

PCPU=[on | off]

Physical CPU (PCPU) で始まるメトリックでしかない、PCPU セクションの記録を使用可能または使用不可にします。これらのメトリックは、Processor Utilization of Resources Register (PURR) に基づいています。デフォルト値は **off** です。

PCPU セクションには、次の値を指定できます。

on PCPU セクションの記録を使用可能にします。

off PCPU セクションの記録を使用不可にします。

SCPU=[on | off]

Scaled CPU (SCPU) で始まるメトリックでしかない、SCPU セクションの記録を使用可能または使用不可にします。これらのメトリックは、Scaled Utilization of Resources Register (SPURR) に基づいています。デフォルト値は **off** です。

SCPU セクションには、次の値を指定できます。

on SCPU セクションの記録を使用可能にします。

off SCPU セクションの記録を使用不可にします。

注: コマンド・ラインで同じ値が複数回使用された場合、**options** パラメーターの最新値がそれ以前の値をオーバーライドします。

例: コマンド "nmon -f -y PCPU=on -y PCPU=off" を実行した場合、PCPU オプションには値 **off** が使用されます。

-Y	追加されて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。 -t 、 -T 、または -Y フラグと一緒に指定できません。
-z	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、 -f -s 900 -c 96 と同じです。
-Z	実行しようとする nmon コマンドの優先順位を指定します。値 -20 は重要であることを意味します。値 20 は重要でないことを意味します。 root ユーザーだけが負の値を指定できます。
-^	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

項目	説明
<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。
<i>seconds</i>	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。
<i>outputpath</i>	出力ファイルのパスを指定します。

サブコマンド

項目	説明
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・チャンネル・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 (AIO サーバー) 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、 -k 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。ディスクの追加の統計を表示するには、 D 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 -g 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	メモリー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。 M 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。 N 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。x、または Ctrl+C 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、システム名、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。

項目	説明
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。 <ul style="list-style-type: none"> • 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。
u	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、 u キーを 2 回押します。
U	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workload partitionworkload partition情報を表示します。
v	定義済みシステム・リソース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。
V	ディスク・ボリューム・グループの統計を表示します。
w	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。
W	ワークロード・マネージャー (WLM) の統計を表示します。
[カスタムのオンデマンド記録を発生させます。開始した記録は、以前に停止されていない場合、対話式 nmon と一緒に終了します。
]] により発生したカスタム記録を停止します。

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。
- 論理区画の省電力モード。
- 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システム ID です。
- 現行のシリアル番号。この番号は、現行のシステム ID、または動的構成イベント以降の区画のシステム 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。
- 論理区画のサブプロセッサ・モード。

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。

項目	説明
Total AIO Processes	AIO プロセスの総数。
Actually in use	プロセッサの 0.1% を超える量を使用している AIO プロセス数。
CPU Used	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。
All time peak	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。
Recent peak	プロセッサの 0.1% を超える量を使用しているカーネル・プロセスの最近の最大数。
Peak	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。

項目	説明
pid	プロセスの ID。
ppid	親プロセスの ID。
User	プロセスのユーザー ID。
Proc Group	プロセス・グループの ID。
Nice	プロセスの初期優先順位。この値は、 nice コマンドにより設定されます。
Priority	プロセスの基本スケジューリング優先順位。
Status	プログラムの状況。
Proc_Flag	プロセスのフラグ。
Thrds	スレッド数。
ファイル	使用中の最大ファイル索引。
Foreground	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。
コマンド	コマンド名。
Time Start	コマンドが開始した時刻。
CPU-Total	プロセスの開始以降にそのプロセスが要した合計時間。
Child Total	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。
Delta-Total	その間隔の中でそのプロセスが要した合計時間。
%CPU Used	最後の間隔の中で使用したプロセッサのパーセンテージ。
Size KB	ページのサイズ (キロバイト単位)。
Res Size	実メモリ・データ (常駐セット) の合計、およびプロセスの実メモリ (常駐セット) 文字サイズの合計。
Res Set	実メモリ・データ (常駐セット) の合計、およびプロセスの実メモリ (常駐セット) 文字サイズの合計。
Res Text	プロセスの実メモリ文字サイズ。
Res Data	プロセスの実メモリ・データ・サイズ。
Char I/O	最後の間隔での入出力文字数/秒。
RAM Use	使用された RAM のパーセンテージ。
Paging I/O	最後の間隔での入出力ページ不在数/秒。
Paging Other	最後の間隔での非入出力ページ不在数/秒。
Paging Repages	最後の間隔での再ページ不在数/秒。
クラス	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、論理プロセッサ、対応ライセンス、および使用された仮想プロセッサの、ユーザー、システム、アイドル、および待ち時間の概要が表示されます。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
- LPAR が SMT にバインドされているか、使用可能にされているか
- LPAR フラグが設定されているかどうか、および AVG=1p より大きな値を表示するように設定されているかどうか

フラグが設定されている場合、**nmon+C** グラフには、それぞれ、Cpu_user および Avg_user に関する情報が示されます。グラフを右側の列に表示できます。

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。

項目	説明
Max Phys in Sys	システム内の物理プロセッサの最大数
Phys CPU in system	システム内の物理プロセッサ数
Virtual Online	オンラインの仮想プロセッサ数
Logical online	オンラインの論理プロセッサ数
Physical pool	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
SMT threads/CPU	SMT スレッド数/プロセッサ

キャパシティー:

以下の情報がプロセッサのキャパシティー情を表示します。

項目	説明
Cap. Processor Min	この LPAR 用に定義された処理単位の最小数
Cap. Processor Max	この LPAR 用に定義された処理単位の最大数
Cap. Increment	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
Cap. Unallocated	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
Cap. Entitled	ライセンス済みキャパシティー
MinReqVirtualCPU	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。

項目	説明
LPAR ID Group:Pool	LPAR グループの ID とそのプール ID
Memory (MB/GB) Min:Max	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)
Memory(MB/GB) Online	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)
Memory Region LMB	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):

項目	説明
Time Dispatch Wheel	各仮想プロセッサがそのライセンスを受け付ける間隔
MaxDispatch Latency	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)
Time Pool Idle	共用プロセッサ・プールがアイドルとなる時間 (秒単位)
Time Total Dispatch	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。

項目	説明
Virtual CPU (Min - Max)	LPAR 定義にある仮想プロセッサの最小数と最大数
Logical CPU (Min - Max)	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。

項目	説明
Weight Variable	プロセッサ・キャパシティーに関する可変の重み付け
Weight Unallocated	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。

項目	説明
Root	NFS V2 サーバーとクライアントのルート要求
Wrcache	NFS サーバーとクライアントの書き込みキャッシュ要求
Null	NFS サーバーとクライアントの書き込みキャッシュ要求
Getattr	NFS サーバーとクライアントの属性獲得要求
Setattr	NFS サーバーとクライアントの属性設定要求
Lookup	NFS サーバーとクライアントのファイル名検索要求
Readlink	NFS サーバーとクライアントのリンク読み取り要求
読み取り	NFS サーバーとクライアントの読み取り要求
Write	NFS サーバーとクライアントの書き込み要求
Create	NFS サーバーとクライアントのファイル作成要求
Mkdir	NFS サーバーとクライアントのディレクトリー作成要求
Symlink	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
Remove	NFS サーバーとクライアントのファイル除去要求
Rmdir	NFS サーバーとクライアントのディレクトリー除去要求
Rename	NFS サーバーとクライアントのファイル名前変更要求

項目	説明
Link	NFS サーバーとクライアントのリンク作成要求
Readdir	NFS サーバーとクライアントのディレクトリー読み取り要求
Fsstat	NFS サーバーとクライアントのファイル状況要求
Access	NFS V3 サーバーとクライアントのアクセス要求
Mknod	NFS V3 サーバーとクライアントの mknod 作成要求
readdir+	NFS V3 サーバーとクライアントのディレクトリー読み取りプラス要求
Fsinfo	NFS V3 サーバーとクライアントのファイル情報要求
Pathconf	NFS V3 サーバーとクライアントのパス構成要求
Commit	NFS サーバーとクライアントのコミット要求
Bad calls	NFS サーバーとクライアントの失敗した呼び出し
Calls	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。

項目	説明
Access	NFS V4 サーバーとクライアントのアクセス要求
acl_read	NFS V4 クライアントの読み取りアクセス制御リスト (ACL)
acl_stat_l	NFS V4 クライアントの長 ACL 情報取得
acl_write	NFS V4 クライアントの書き込みアクセス制御リスト (ACL)
Clntconfirm	NFS V4 クライアントの確認操作
Close	NFS V4 クライアントのファイル・クローズ
Commit	NFS V4 サーバーとクライアントをコミット
Compound	NFS V4 サーバーの複合呼び出し
Create	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
Deletpurge	NFS V4 サーバーが代行待ちリカバリーを消去
Deletpurge	NFS V4 サーバーとクライアントの代行戻し
Finfo	ファイル情報を取得している NFS V4 クライアント
getattr	NFS V4 サーバーとクライアントの属性取得
getfh	NFS V4 サーバーのファイル・ハンドル取得
Link	操作をリンクしている NFS V4 サーバーとクライアント
Lock	操作をロックしている NFS V4 サーバーとクライアント
lockt/test	指定されたロックをテストしている NFS V4 サーバー、または NFS V4 クライアント・ロック・テスト
locku/unlock	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
lookup	ファイル名を検索している NFS V4 サーバーとクライアント
lookupp	親ディレクトリーを検索している NFS V4 サーバー
mkdir	ディレクトリーを作成している NFS V4 クライアント
mknod	特殊ファイルを作成している NFS V4 クライアント
Null	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
nverify	NFS V4 サーバーの属性内の相違検査
openattr	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
openconfirm	使用のためにオープンを確認している NFS V4 サーバーとクライアント
opendowngrade	指定ファイルのアクセス権限を格下げしている NFS V4 サーバーとクライアント
Open	NFS V4 サーバーとクライアントのオープン操作
操作	NFS V4 サーバーとクライアントの操作
pcl_read	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
pcl_readstat_l	NFS V4 クライアントの pcl_stat 詳細操作
pcl_stat	NFS V4 クライアントの pcl_stat 操作
pcl_write	NFS V4 クライアントの pcl_write 操作
putfh	NFS V4 サーバーの現行ファイル・ハンドル設定
putpubfh	NFS V4 サーバーの一般ファイル・ハンドル設定
putrootfh	NFS V4 サーバーのルート・ファイル・ハンドル設定
readdir	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
readlink	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
読み取り	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
release	NFS V4 サーバーとクライアントの release_lock 操作
削除	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

項目	説明
rename	NFS V4 サーバーとクライアントのオブジェクト名変更
renew	NFS V4 サーバーとクライアントのリース更新
replicate	NFS V4 クライアントの複製操作
restorefh	NFS V4 サーバーのファイル・ハンドル復元
rmdir	NFS V4 クライアントのディレクトリー除去
savefh	NFS V4 サーバーのファイル・ハンドル保存
secinfo	NFS V4 サーバーとクライアントがセキュリティー情報を入手
setattr	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
setclient	NFS V4 サーバーとクライアントの setclient 操作
statfs	NFS V4 クライアントのファイル統計情報要求
symlink	NFS V4 クライアントのシンボリック・リンク操作
verify	NFS V4 クライアントが同じ属性を検査
write	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。

項目	説明
I/F Name	インターフェース名
Recv-KB/s	この間隔の中で受信されたデータ (キロバイト/秒)
Trans-KB/s	この間隔の中で送信されたデータ (キロバイト/秒)
Packin	この間隔の中で受信されたパケット数
Packout	この間隔の中で送信されたパケット数
Insize	この間隔の中で受信された平均パケット・サイズ
Outsize	最後の間隔の中で送信された平均パケット・サイズ
Peak->Recv	受信データのピーク値 (キロバイト/秒)
Peak->Trans	送信データのピーク値 (キロバイト/秒)
Total Recv	合計受信データ (メガバイト/秒)
Total Sent	合計送信データ (メガバイト/秒)
MTU	トランスポート・ユニットの最大サイズ (バイト単位)
Ierror	入力エラー数
Oerror	出力エラー数
Collision	衝突回数
Mbits/s	アダプター・ビット・レート (メガビット/秒)。ネットワーク・アダプターの転送量が 10Gb を超える場合は、アダプター・ビット・レートは 10240 メガビット/秒で示されます。
説明	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。

項目	説明
CPU	そのクラスのプロセッサ使用のパーセンテージ
MEM	そのクラスの物理メモリー使用のパーセンテージ
BIO	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
Process (Procs)	そのクラス内のプロセス数
Tier (T)	層番号。この値の範囲は 0 から 9 です。
Inheritance (I)	継承属性の値。0 の値は、no を表し、1 の値は yes を表します。
位置	ロケーションの値。1 の値の意味は、共用クラス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
O	50 未満
O	60 未満
8	70 未満
X	80 未満
#	90 未満
@	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。

項目	説明
Name	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。
ディスク	グループにあるディスク数。
Read/Write-KB/s	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)
TotalMB/s	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)
Xfers/s	この間隔の中での読み取りと書き込みのデータ転送回数
BlockSizeKB	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。

項目	説明
Name	仮想バスの名前。
サイズ	ESS バスのサイズ。
AvgBusy	このディスクの平均のビジー使用。
Write-KB/s	この間隔の中での書き込みデータ転送率 (キロバイト/秒)
Read-KB/s	この間隔の中での読み取りデータ転送率 (キロバイト/秒)
Xfers/s	読み取りおよび書き込みの転送回数/秒
Total vpaths	仮想バス数。

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。

項目	説明
FileSystem	ファイルシステム名。
Size (MB)	ファイルシステムのサイズ (メガバイト)。
Free (MB)	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)。
%Used	使用済みのファイルシステムのパーセント。
%Inodes	i-node で使用されているファイルシステムのパーセント。
Mount point	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。

項目	説明
runqueue	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。
pswitch	この間隔の中でのプロセッサ・スイッチ回数/秒。
fork	この間隔の中での fork の回数/秒。
exec	この間隔の中での exec の回数/秒。
msg	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
sem	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。
hw intrp	この間隔の中でのデバイス割り込みの回数/秒。
sw intrp	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。
Swapin	この間隔の中でスワップ・キュー内のプロセス数/秒。
Syscall	この間隔の中でのシステム呼び出し回数/秒。
read	この間隔の中での読み取り呼び出し回数/秒。
write	この間隔の中での書き込み呼び出し回数/秒。
readch	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。
Writch	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。
R + W (MB/s)	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)
Uptime	このシステムが稼働状態にあった時間。
iget	この間隔の中での inode ルックアップ回数/秒。
dirblk	この間隔の中で、ファイルのエントリ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。
namei	この間隔の中でのパス名からの vnode 検索回数/秒。
ksched	この間隔の中でのカーネル・プロセス作成回数/秒。
koverf	この間隔の中で、ユーザーが上限まで fork したかプロセスの構成限界まで達した、カーネル・プロセス作成試行回数/秒。
kexit	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。

項目	説明
EntitledCPU	区画のライセンス済みキャパシティー。
UsedCPU	その区画で消費された物理プロセッサ数。

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。

項目	説明
Count	ラージ・ページ数およびその合計サイズ。
Free	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。
In Use	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。
サイズ	ラージ・ページのサイズ。
High water mark	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。

項目	説明
PagingSpace	ページ・スペース数。
Volume-Group	ボリューム・グループ数。
Type	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。
LPs	論理区画サイズ。
MB	サイズ (メガバイト単位)。
Used	ボリューム・グループの使用パーセンテージ。
IOpending	ページング・スペース内の保留入出力数。
Active/Inactive	アクティブまたは非アクティブのページング・スペース。
Auto/NotAuto	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。

項目	説明
Name	ボリューム・グループ名。
ディスク	グループにあるディスク数。
AvgBusy	ボリューム・グループにあるディスクの平均ビジー。
Read/Write-KB/s	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)
TotalMB/s	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)
Xfers/s	この間隔の中での読み取りと書き込みの転送回数
BlockSizeKB	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。

項目	説明
Name	ディスクの名前。
Busy	そのディスクの平均ビジー。
Read-KB/s	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)
Write-KB/s	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)
Transfers/sec	この間隔の中での読み取りと書き込みの転送回数
SizeKB	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)
Peak	平均ビジーのピーク・パーセンテージ。
Peak KB/s	データの読み取りと書き込みのピーク (キロバイト/秒)。
qDepth	ディスクへの送信要求数で、完了していない送信要求数。
Totals Size (GB)	ディスクの合計サイズ (ギガバイト)。
Totals Free (GB)	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)。
Totals Read (MB/s)	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)
Totals Write (MB/s)	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。

項目	説明
Name	ディスク名。
Size (GB)	ディスク・サイズ (ギガバイト)。
Free (GB)	ディスクに残っているフリー・スペース (ギガバイト)。
Disk Paths	ディスクに定義されたパス数。
Disk Adapter	ディスク・アダプター名。
Volume Group	ディスクが所属するボリューム・グループ。
Disk Description	ディスクの記述。
Totals Size (GB)	ディスクの合計サイズ (ギガバイト)。
Totals Free (GB)	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)。
Totals Read (MB/s)	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)
Totals Write (MB/s)	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。

項目	説明
Disk	ディスク名
Service (in msecs)	平均サービス時間/要求 (ミリ秒)。
Wait (in msecs)	平均待ち時間/要求 (ミリ秒)。
ServQ size	サービス・キュー内の平均要求数。
WaitQ size	完了待ちになっている平均要求数。
ServQ Full	入ってくる要求をそのディスクが受け入れようとしな回数。
Totals Size (GB)	ディスクの合計サイズ (ギガバイト)。
Totals Free (GB)	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)。
Totals Read (MB/s)	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)
Totals Write (MB/s)	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)

グラフ付きディスク統計 (D キーを 4 回押す)

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。

項目	説明
Name	ディスク名
Busy	このディスクの平均ビジー・パーセンテージ。
Read-KB/s	読み取りのデータ転送速度 (キロバイト/秒)
Write-KB/s	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。

項目	説明
%Used	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。
%Free	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。
MB Used	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)。
MB Free	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)。
Pages/sec to Paging Space	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。
Pages/sec to file system	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。
Page Scans	クロックによるページ・スキャン回数。
Page Faults	ページ・フォールト数。
Page Cycles	ページ置換サイクル数。
Page Steals	ページ・スチール数。
Numperm	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)。
Process	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。
System	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。
Free	空き状態の実メモリーのパーセンテージ。
Total	使用された合計の実メモリーのパーセンテージ。
Min/Maxperm	ページ・スチールに対する minperm と maxperm 値。
Min/Maxfree	minfree と maxfree のページ・フリー・リスト。
Min/Maxpgahead	ページ・アヘッドのページの最小数と最大数。
Total Virtual	合計仮想メモリー。
Accessed Virtual	活動状態の仮想メモリー。
Numclient	クライアント・フレーム数。
Maxclient	クライアント・フレームの最大数。
User	非システム・セグメントが使用する実メモリー。
Pinned	ピンされた実メモリー。

AMS 統計は、**topas_nmon** メモリー・パネルに表示されます。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。

項目	説明
Pool	論理区画 (LPAR) が属するプールの AMS プール ID。
重み付け	可変メモリーの重み付け。
pMem	論理メモリー区画を現在バックアップしている物理メモリー (MB)。
hpi	ハイパーバイザーのページインの回数。
hpit	ハイパーバイザーのページインにかかる時間 (秒単位)。

論理装置の情報:

項目	説明
Size (MB)	論理装置に割り当てられた合計サイズ。
Lu Udid	論理装置 ID。

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。

項目	説明
Adapter	アダプター名。
Busy%	アダプターの帯域幅使用。これは、このアダプターに接続されたディスクの集合 Busy% です。アダプターに複数のディスクが接続されている場合は、この値が 100% を超えることがあります。
Read-KB/s	読み取りのデータ転送速度 (キロバイト/秒)
Write-KB/s	書き込みのデータ転送速度 (キロバイト/秒)
Transfers	読み取りおよび書き込みの転送回数
ディスク	ディスク数。
Adapter-Type	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。

項目	説明
Number	シリアル番号。
Name	共用イーサネット・アダプターの名前。
Recv-KB/s	受信されたデータのデータ転送速度 (キロバイト/秒)
Trans-KB/s	送信されたデータのデータ転送速度 (キロバイト/秒)
Packin	この間隔の中での受信パケット数/秒。
Packout	この間隔の中での送信パケット数/秒。
Insize	この間隔の中で着信したパケットの平均サイズ/秒。
Outsize	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、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。

項目	説明
Numframes	ページ・サイズの実メモリー・フレーム数。
Numfrb	フリー・リスト上のページ数。
Numclient	クライアント・フレーム数。
Numcompress	圧縮セグメント内のフレーム数。
Numperm	非作業用セグメント内のフレーム数。
Numvpages	アクセスされた仮想ページ数。
Minfree	最小フリー・リスト。
Maxfree	最大フリー・リスト。
Numpout	ページ・アウト回数。
Numremote	リモート・ページ・アウト回数。
Numwseguse	作業用セグメントとして使用中のページ数。
Numpseguse	永続セグメントとして使用中のページ数。
Numclseguse	クライアント・セグメントとして使用中のページ数。
Numwsegin	作業用セグメントとしてピンされたページ数。
Numpsegin	永続セグメントとしてピンされたページ数。
Numclsegin	クライアント・セグメントとしてピンされたページ数。
numpgsp_pgs	割り当てられたページ・スペース数。
numralloc	リモート割り当て数。
pfrsvdblks	システム予約済みブロック数。
Pfavail	ピン用に使用可能なページ数。
Pfpinavail	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。
system_pgs	V_SYSTEM 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。
nonsys_pgs	V_SYSTEM 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。
Numpermio	非作業用記憶域内のページ・アウト数。
Pgexct	ページ・フォールト数。
Pgrclm	ページ再利用数。
Pageins	ページインされたページ数。
Pageouts	ページアウトされたページ数。
Pgspgins	ページ・スペースからページインされたページ数。
Pgspgouts	ページ・スペースからページアウトされたページ数。
Numsios	開始された入出力回数。
Numiodone	完了した入出力回数。
Zerofills	ゼロで埋め込まれたページ数。
Exfills	exec で埋め込まれたページ数。
Scans	クロックによるページ・スキャン回数。
Cycles	クロック・ハンド・サイクル数。
pgsteals	ページ・スチール数。

ファイバー・チャンネル・アダプター統計

このビューには、ファイバー・チャンネル・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。

項目	説明
Number	シリアル番号。
Name	ファイバー・チャンネル・アダプターの名前。
Receive-KB/s	受信されたデータのデータ転送速度 (キロバイト/秒)
Transmit-KB/s	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。

スレッド・レベルの統計

このビューには、スレッド・レベル統計に関する情報が含まれます。このビューを表示するには、**-i** 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。

項目	説明
PID	スレッドが属するプロセス ID。
TID	より高い CPU を使用するトップ・スレッド ID。ソートは降順での CPU 使用率に基づいています。
%CPU	特定のスレッドによって使用される CPU のパーセンテージ。
BOUND CPU ID	スレッドがプロセッサにバインドされている場合は、バインドされた CPU ID。

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。

項目	説明
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:nmon:topas コマンドを実行できます。
NMON	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。
NMON_TIMESTAMP	NMON_TIMESTAMP 変数は、次の値に指定できます。 NMON_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。 NMON_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 記録開始時に始動される外部コマンド。
NMON_END	nmon 記録終了時に始動される外部コマンド。
NMON_SNAP	メトリックを記録するために定期的に始動される外部コマンド。
NMON_ONE_IN	以下の値に合わせて、 NMON_ONE_IN 変数を指定できます。 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 <i>n</i> 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

1. 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
```

2. **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。

- a. 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. 以下の例に示したとおりに、環境変数を設定します。

```
$export NMON_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
```

生成された nmon ファイルを、外部記録処理によって生成される ps.csv ファイルに連結するには、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。

8. **hdisk** の詳細を表示するには、**-k** フラグを指定して **nmon** コマンドを入力します。

```
nmon -k hdisk1,hdisk2
```

このコマンドは、**hdisk1** および **hdisk2** のディスク詳細を表示します。**hdiskpower** デバイスの場合には、以下のコマンドを入力します。

```
nmon -k hdiskpower or
nmon -k power
```

注: **nmon -k hdisk** は、LPAR 上のすべての **hdisk** デバイスと一致し、**hdiskpower** デバイスとは一致しません。

hdiskpower デバイスはすべて、対話式および記録モードで **power** として表示されます。例えば、**nmon -k hdiskpower1** はデバイス **hdiskpower1** と一致し、**nmon -k hdiskpower** は LPAR 上のすべての **hdiskpower** デバイスと一致します。

注: **nmon** 記録ファイル内の **lscnf** コマンドおよび **lspv** コマンドの出力は、**nmon-k** コマンドの変更によって影響を受けません。

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

ネットワークのチューニング・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**] [**-F**] **-a**

no -h [*Tunable*]

no -F -L [*Tunable*]

no -F -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、ネットワークのチューニングに使用される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システム・ブート値を設定または表示します。このコマンドを使用して、永続的な変更を行うことも、次のシステム・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、両方のアクションを実行します。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・パラメーターがどのように相互作用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管理」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、『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読んでから、その目的を理解するために『参照』で示される指示に従ってください。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、この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』と『チューニング』の両方のセクションに「N/A」と記されている場合は、AIX の開発側から指示がない限り、このパラメーターを変更してはなりません。

フラグ

項目	説明
-a	すべてのチューナブル・パラメーターの現行値、リブート値 (-r と併用した場合)、または永続値 (-p と併用した場合) を、 Tunable = Value を対にして、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ブート値と現行値が等しい場合にのみ表示されます。そうでない場合は値として NONE が表示されます。

項目

-d *Tunable*

説明

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 (*Tunable* が以下のいずれかの値に設定されていて、

- チューナブルがデフォルト値に設定されておらず、タイプ *Bosboot* または *reboot* である
- チューナブルがタイプ *Incremental* であり、デフォルト値から変更する必要がある

かつ、**-r** と組み合わせて使用されていないとき)、チューナブル・パラメーターは変更されませんが、警告メッセージが表示されます。

-D

すべてのチューナブル・パラメーターをデフォルト値にリセットします。変更する必要があるチューナブル・パラメーターが以下のいずれかのタイプであり、

- *Bosboot* または *Reboot*
- タイプ *Incremental* であり、デフォルト値から変更されている

かつ、**-p** と **-r** のどちらのフラグとも組み合わせて使用されていない場合、パラメーターは変更されませんが、警告メッセージが表示されます。コマンド・ラインでオプション **-a**、**-L**、または **-x** を指定したとき、制限付きチューナブル・パラメーターの表示を強制します。 **-F** フラグを指定しない場合、表示オプションとの関連で制限付きチューナブルが指定されない限り、これらのチューナブルは含まれません。

-F

-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 パラメーター							

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

項目	説明
-o <i>Tunable</i> [= <i>NewValue</i>]	<p><i>Tunable</i> の値を表示するかまたは <i>NewValue</i> の値に設定します。チューナブルを変更する必要がある場合、つまり指定値が現行値と異なり、以下のいずれかのタイプであり、</p> <ul style="list-style-type: none"> • Bosboot または Reboot • Incremental であり、現行値が指定値より大きい <p>かつ、-r と組み合わせて使用されない場合、変更されませんが、警告メッセージが表示されます。</p>
-p	<p>新しい値を指定せずに -r を使用すると、<i>Tunable</i> 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、<i>tunable</i> 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> <p>-o、-d、または -D と組み合わせて使用された場合、現行値とリブート値の両方に変更が適用されます。つまり、現行値の更新だけでなく、<i>/etc/tunables/nextboot</i> 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。</p>
-r	<p>新しい値を指定せずに -a または -o と一緒に使用されると、パラメーターの現行値と次のブート値が等しいときには値が表示されます。そうでない場合は値として NONE が表示されます。</p> <p>-o、-d、または -D フラグと組み合わせて使用されると、リブート値に変更が適用されます。つまり、<i>/etc/tunables/nextboot</i> 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。</p>
-x [<i>Tunable</i>]	<p>1 つまたはすべての <i>Tunable</i> 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。</p> <pre>tunable,current,default,reboot,min,max,unit,type,{dtunable }</pre> <p>where:</p> <pre>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</pre>

制限付きチューナブル・パラメーターに対して **-o**、**-d**、または **-D** フラグを使用して変更すると、限定使用タイプのチューナブル・パラメーターが変更されたという警告メッセージが表示されます。コマンド・ラインで **-r** または **-p** オプションも指定した場合は、変更の確認を求めるプロンプトが出されます。システムのリブート時に、コマンド・ラインを使用し、**-r** または **-p** オプションを指定することによってデフォルト値とは異なる値に変更されていた制限付きチューナブルが */etc/tunables/nextboot* ファイルに存在すると、それらの変更されたチューナブルのリストを示すエラー・ログ・エントリーが作成されます。

タイプ Mount のパラメーターに対して **-o**、**-d**、または **-D** フラグを使用することによって変更した場合は、その変更が将来のマウントに有効であるという警告メッセージが表示されます。

-o、**-d**、または **-D** フラグを使用することによってタイプ Connect のパラメーターを変更すると、**inetd** が開始され、その変更が将来のソケット接続に有効であるという警告メッセージが表示されます。

-o、**-d**、または **-D** フラグを使用し、**-r** フラグを使用せずにタイプ Bosboot または Reboot のパラメーターを変更すると、エラー・メッセージが表示されます。

-o、**-d**、または **-D** フラグを使用し、**-r** フラグを使用せずに、現行値より小さい新しい値を指定してタイプ Incremental のパラメーターの現行値を変更すると、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** コマンドなど) 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。

項目	説明
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変更できる場合
Bosboot	bosboot を実行してマシンをリポートする場合に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	このパラメーターを変更できず、このパラメーターが 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** コマンドを実行するかどうかを尋ねるプロンプトをユーザーに出します。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。

注: **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。

チューナブル・パラメーター

チューナブルのデフォルト値および値の範囲については、**no** コマンド・ヘルプ (**-h** <チューナブル・パラメーター名 >) を参照してください。

項目	説明
arpqsize	目的: アドレス解決プロトコル (ARP) の応答を待機している間にキューに入れられるパケットの最大数を指定します。 チューニング: この属性は、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。
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。 チューニング: 安定したネットワークで ARP アクティビティを減らすには、 arpt_killc を増やします。

項目 arptab_bsiz	<p>説明</p> <p>目的: アドレス解決プロトコル (ARP) テーブルのバケット・サイズを指定します。</p> <p>チューニング:</p> <p>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。多数のエントリーが消去される場合は、ARP テーブルのサイズを大きくする必要があります。ARP テーブルのハッシュ分布を表示するには、arp -a を使用します。</p>
arptab_nb	<p>目的: ARP テーブルのバケットの数を指定します。</p> <p>チューニング:</p> <p>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。多数のエントリーが消去される場合は、ARP テーブルのサイズを大きくする必要があります。ARP テーブルのハッシュ分布を表示するには、arp -a を使用します。多数のクライアントまたはサーバーがあるシステムの場合は、この値を大きくします。デフォルトでは $149 \times 7 = 1043$ の 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。</p>
bcasting	<p>目的: ブロードキャスト・アドレスに対する ICMP エコー・パケットへの応答を可能にします。</p> <p>チューニング:</p> <p>値を 0 にするとこれが使用不可になり、1 にするとこれが使用可能になります。デフォルトでは、ブロードキャスト・アドレスに対するエコー・パケットに応答しません。これにより、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」が防止されます。</p>
clean_partial_conns	<p>目的: SYN のアタックを回避するかどうかを指定します。ゼロ以外の場合、clean_partial_conns は、アタックでない新しい接続用のスペースを確保するためにランダムに除去する部分接続の数を指定します。</p> <p>チューニング:</p> <p>値を 0 にすると、このオプションが無効になります。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があるサーバーに対してオンにしてください。</p>
delayack	<p>目的: ある特定の 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の結合を試行します。</p> <p>チューニング:</p> <p>このアクションは、宛先ポートが delayackports 属性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ため、HTTP サーバーとの通信時にパフォーマンスが向上します。このパラメーターには、次の 4 つの値のいずれかを指定できます。</p> <ul style="list-style-type: none"> 0 遅らせない、通常の操作 1 サーバーの SYN の ACK を遅らせる 2 サーバーの FIN の ACK を遅らせる 3 SYN と FIN の両方の ACK を遅らせる

項目	説明
delayackports	<p>目的: <code>delayack port</code> オプションで定義した操作が実行される宛先ポートのリストを指定します。</p> <p>チューニング: この属性には最大 10 個のポートを指定できます。各ポートはコンマで区切り、中括弧で囲みます。以下に例を示します。</p> <pre>no -o delayackports={80,30080}.</pre> <p>リストをクリアするには、オプションを {} に設定します。</p>
dgd_flush_cached_route	<p>目的: デッド・ゲートウェイ検出により、前のデッド・ゲートウェイがオンラインに戻されたことが検出されたら、ソケットのキャッシュ経路をフラッシュします。データが送信される前に、接続は経路の再取得を強制されます。</p> <p>チューニング: 値が 1 の場合、DGD によるキャッシュ経路のフラッシュが使用可能になります。値が 0 だと、これが使用不可になります。</p>
dgd_packets_lost	<p>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断するまでに消失させる必要がある連続パケットの数を指定します。</p>
dgd_ping_time	<p>目的: アクティブ・デッド・ゲートウェイ検出による、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。</p>
dgd_retry_time	<p>目的: 経路のコストがパッシブ・デッド・ゲートウェイ検出によって発生したときに、それを発生したままにしておく必要がある時間 (分単位) を指定します。この分数が経過すると、経路のコストはユーザーが構成した値に復元されます。指定される単位は数値です。</p>
directed_broadcast	<p>目的: ゲートウェイへのダイレクテッド・ブロードキャストを許可する必要があるかどうかを指定します。</p> <p>チューニング: 値が 1 の場合、ゲートウェイの相手側にあるネットワーク上でブロードキャストされる必要があるパケットをゲートウェイに送信することができます。</p>
fasttimo	<p>目的: TCP 高速タイムアウト・タイマーにミリ秒の遅延を設定できます。このタイムアウトは、遅らせた肯定応答を送信するために、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。</p> <p>チューニング: IBM 以外のシステムによっては、このタイマー値を小さくするとパフォーマンスが向上するものがあります。ただし、このパラメーターにより、システム使用率がわずかに増加することがあります。</p>
hstcp	<p>目的: RFC 3649 で指定したように HighSpeed TCP を使用可能にします。このパラメーターは、大規模輻輳 (ふくそう) ウィンドウとの TCP 接続で使用する輻輳制御メカニズムを変更して、平均スループットを改善します。</p> <p>チューニング: 値が 1 だと、HighSpeed TCP 機能拡張がシステム全体で使用可能になります。値が 0 だと、これが使用不可になります。</p>
icmp6_errmsg_rate	<p>目的: 1 秒につき送信できる ICMP v6 エラー・メッセージの数の上限を指定します。このパラメーターは、ICMP v6 エラー・メッセージによる帯域幅の過剰使用を防止します。</p>
icmpaddressmask	<p>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。</p> <p>チューニング: 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。</p>

項目	説明
ie5_old_multicast_mapping	<p>目的: 値 1 が使用された場合に、機能アドレスではなくブロードキャスト・アドレスにマップされる必要があるトークンリング上の IP マルチキャストを指定します。</p>
ifsize	<p>目的: 単一タイプ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。この制限は、イーサネット・インターフェース構造をいくつでも扱えるようにインフラストラクチャーが動的に拡張されるイーサネット・インターフェース構造には適用されません。</p> <p>チューニング: ホット・プラグ・アダプターをサポートするシステムおよび DLPAR 構成では、アダプターを必要に応じて追加できるため、ifsize パラメーターを大きくする必要があります。このシステムまたはパーティション用に追加される多数のアダプターを受け入れるために、静的インターフェース・テーブルは十分に大きくなければなりません。システムは、始動時に、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、存在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。</p>
ip6_defttl	<p>目的: 他のホップ・カウントが指定されていない場合、IP バージョン 6 の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。</p>
ip6_prune	<p>目的: IP バージョン 6 の経路指定テーブルの有効期限切れ経路を検査する頻度を秒数で指定します。</p>
ip6forwarding	<p>目的: カーネルが IP バージョン 6 のパケットを転送する必要があるかどうかを指定します。</p> <p>チューニング: デフォルト値の 0 では、ipv6 パケットがローカル・システム用でない場合は転送されません。1 の値では転送が使用可能になります。</p>
ip6srcrouteforward	<p>目的: システムが送信元で経路指定された IP バージョン 6 のパケットを転送するかどうかを指定します。</p> <p>チューニング: 値を 1 にすると、送信元で経路指定されたパケットを転送できます。0 の値では、送信元で経路指定されて、その宛先がないすべてのパケットが廃棄されます。</p>
ip_ifdelete_notify	<p>目的: インターフェース・アドレスが削除される場合を指定します。インターフェース・アドレスにローカルにバインドされ、削除された既存の TCP 接続はすべて、エラー ENETDOWN で通知される必要があります。</p> <p>チューニング: ENETDOWN エラーが戻された場合、既存の FTP/Telnet 接続が切断されます。</p>
ip_nfrag	<p>目的: 一度に IP 再組み立てキューに保持できる IP パケットのフラグメントの最大数を指定します。</p>
ipforwarding	<p>目的: カーネルがパケットを転送する必要があるかどうかを指定します。</p> <p>チューニング: システムが IP ルーターとして動作している場合は、このパラメーターを 1 に設定します。</p>
ipfragttl	<p>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。</p> <p>チューニング: タイムアウト後にドロップされたフラグメントを検査します (netstat -p ip)。値 IP、つまりタイムアウト後にドロップされたフラグメントがゼロ以外であるために ipfragttl パラメーターが大きくなる場合、再送の件数が減る可能性があります。</p>
ipignoreredirects	<p>目的: 受信された宛先変更を処理するかどうかを指定します。</p> <p>チューニング: 値が 0 の場合は、宛先変更が通常通りに処理されます。1 の値は宛先変更を無視します。</p>

項目 ipqmaxlen	<p>説明</p> <p>目的: IP プロトコル入力キューで待機させることができる受信パケットの数を指定します。</p> <p>チューニング: ipintrq のオーバーフローを検査するか (<code>netstat -s</code>)、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・カウンターにアクセスします。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、サイズを大きくします。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。これらのデバイスで ipqmaxlen パラメーターを大きくしても効果はありません。</p>
ipoutqueues	<p>目的 IPv を介して送信される User Datagram Protocol (UDP) パケットをキューに入れるかどうかを指定します。これらの UDP パケットは、個別のカーネル・スレッドによって扱われます。</p> <p>チューニング: デフォルト値は 0 であり、UDP がパケットをキューに入れずに即時に送信することを指定します。ゼロ以外の値は、作成および使用されるキューの数を指定します。例えば、UDP が使用する単一のキューを作成する場合は、以下のコマンドを入力します。</p> <pre>no -o ipoutqueues=1</pre>
ipsendredirects	<p>目的: カーネルが宛先変更シグナルを送信する必要があるかどうかを指定します。</p> <p>チューニング: このパラメーターは、パフォーマンス結果を伴う構成上の決定です。</p>
ipsrcrouteforward	<p>目的: システムが送信元で経路指定されたパケットを転送するかどうかを指定します。</p> <p>チューニング: デフォルト値の 1 によって、送信元で経路指定されたパケットを転送できます。0 の値では、送信元で経路指定されて、その宛先にないすべてのパケットが廃棄されます。</p>
ipsrcrouterrecv	<p>目的: システムが送信元で経路指定されたパケットを受け入れるかどうかを指定します。</p> <p>チューニング: デフォルト値の 0 では、送信元で経路指定されて、このシステムが宛先になっているすべてのパケットが廃棄されます。1 の値では、送信元で経路指定されたパケットを受け取れます。</p>
ipsrcroutesend	<p>目的: アプリケーションが送信元で経路指定されたパケットを送信できるかどうかを指定します。</p> <p>チューニング: デフォルト値の 1 によって、送信元で経路指定されたパケットを送信できます。0 の値では、アプリケーションが送信元経路指定オプションを設定しようとするとき setsockopt(0) はエラーを戻し、すべての送信元経路指定オプションが発信パケットから除去されます。</p>
limited_ss	<p>目的: RFC 3742 で指定したように Limited SlowStart を使用可能にします。これは、スロー・スタート時に 1 つのウィンドウに対して輻輳ウィンドウをどれだけ拡大させるかのセグメント数を制限します。この機能拡張により、大規模輻輳ウィンドウとの TCP 接続のパフォーマンスが改善されます。</p> <p>チューニング: 1 から 100 までの値にすると、Limited SlowStart の機能拡張がシステム全体で使用可能になり、その値を SlowStart の最大しきい値に対するセグメント数として設定します。値が 0 だと、これが使用不可になります。デフォルト値は 0 です。</p>

項目	説明
llsleep_timeout	<p>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)。</p>
lo_perf	<p>目的: ループバック・パフォーマンスを向上させるために CPU ごとに個別のキューを使用するかどうかを指定します。</p> <p>チューニング: 値が 1 の場合は、CPU ごとに個別のキューを使用できます。値を 0 にすると、このオプションが無効になります。</p>
lowthresh	<p>目的: BPRI_LO 優先順位の allocb 呼び出しを使用して割り当てられる最大バイト数を指定します。</p> <p>チューニング: net_malloc 呼び出しで割り当てられたメモリの合計量がこのしきい値に達すると、BPRI_LO 優先順位の allocb 要求により 0 が戻されます。lowthresh 属性は thewall 属性のパーセントを表し、この値は 0 から 100 まで設定できます。</p>
main_if6	<p>目的: リンクのローカル・アドレスに使用するインターフェースを指定します。</p>
main_site6	<p>目的: サイトのローカル・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。</p>
maxnip6q	<p>目的: IP バージョン 6 のパケットの再組み立てキューの最大数を指定します。</p>
maxttl	<p>目的: RIP パケットの存続時間 (秒単位) を指定します。</p>
medthresh	<p>目的: BPRI_MED 優先順位の allocb 呼び出しを使用して割り当てられる最大バイト数を指定します。</p> <p>チューニング: net_malloc 呼び出しで割り当てられたメモリの合計量がこのしきい値に達すると、BPRI_MED 優先順位の allocb 要求により 0 が戻されます。medthresh 属性は thewall 属性のパーセントを表します。標準的設定値の 95 は、thewall 属性の 95% を表します。</p>
mpr_policy	<p>目的: Multipath Routing に使用されるポリシーを指定します。</p> <p>チューニング: 使用可能な経路指定ポリシーは次のとおりです。</p> <p>重み付けラウンドロビン (1) (route コマンドを使用して) 多重経路に割り当てられたユーザー構成の重みに基づいて、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、これは単純なラウンドロビンと同様に動作します。</p> <p>ランダム (2) 経路をランダムに選択します。</p> <p>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に基づい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出します。個々の重みはそれぞれ、重みの合計から、数値がゼロになるまで除去されます。これにより、選択可能な経路の合計数の範囲内の経路が選出されます。</p> <p>最低使用率 (4) 通過する現行接続の数が最も少ない経路を選択します。</p> <p>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して経路を選択します。</p>

項目 multi_homed	<p>説明</p> <p>目的: マルチホーム IP バージョン 6 ホスト・サポートのレベルを指定します。</p> <p>チューニング:</p> <p>チューニングは、宛先ポートが <code>delayackports</code> パラメーターのリストで指定されている接続に対して実行されます。このパラメーターは、HTTP サーバーとの通信時にパフォーマンスを向上させるために使用できます。このパラメーターには、次の 4 つの値のいずれかを指定できます。</p> <ul style="list-style-type: none"> 0 AIX 4.3 の元の機能を示します。 1 リンク・ローカル・アドレスがそのリンク・ローカル・アドレスの各インターフェースに照会することで解決されることを示します。 2 <code>main_if6</code> で定義されたインターフェースに対してリンク・ローカル・アドレスが検査されることを示します。 3 <code>main_if6</code> で定義されたインターフェースに対してリンク・ローカル・アドレスが検査され、サイト・ローカル・アドレスが <code>main_site6</code> インターフェースに経路指定されることを示します。
nbc_limit	<p>目的: ネットワーク・バッファ・キャッシュに使用できるメモリの合計最大量を指定します。</p> <p>チューニング:</p> <p>この属性はキロバイト数です。キャッシュがこの制限まで達すると、新しいオブジェクトのスペースを確保するために使用頻度の低いキャッシュ・オブジェクトがキャッシュからフラッシュされます。</p>
nbc_max_cache	<p>目的: 専用セグメントを使用せずに、ネットワーク・バッファ・キャッシュで許容されるキャッシュ・オブジェクトの最大サイズを指定します。</p> <p>チューニング:</p> <p>このパラメーターはバイト数です。このサイズより大きいデータ・オブジェクトは、専用セグメントにキャッシュされるか、まったくキャッシュされません。</p>
nbc_min_cache	<p>目的: ネットワーク・バッファ・キャッシュで許容されるキャッシュ・オブジェクトの最小サイズを指定します。</p> <p>チューニング:</p> <p>この属性はバイト数です。このサイズより小さいデータ・オブジェクトは、NBC には書き込まれません。この属性は、<code>send_file()</code> API およびカーネル内の <code>get</code> エンジンを使用するある種の Web サーバーに適用されます。</p>
nbc_ofile_hashsz	<p>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・テーブルのサイズを指定します。</p> <p>チューニング:</p> <p>ハッシュ・テーブル・サイズは、オープンされたファイル・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す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・サイズを変更し、すべての既存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この属性を変更できます。</p>

項目	説明
nbc_pseg	<p>目的: ネットワーク・バッファ・キャッシュ用に作成できる専用セグメントの最大数を指定します。</p> <p>チューニング: このオプションが 0 以外に設定されている場合は、nbc_max_cache に指定されたサイズとセグメント・サイズ (256 MB) の間のデータ・オブジェクトが専用セグメント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・オブジェクトはキャッシュされません。最大数の専用セグメントが存在する場合、専用セグメントの数が制限を超えないように、専用セグメント内のキャッシュ・データが新しいキャッシュ・データのためにフラッシュされることがあります。nbc_pseg が 0 に設定されている場合は、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。</p>
nbc_pseg_limit	<p>目的: ネットワーク・バッファ・キャッシュ内の専用セグメントで許容される最大キャッシュ・データ・サイズを指定します。</p> <p>チューニング: この値はキロバイト数で表されます。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、nbc_pseg_limit 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を制御します。キャッシュ・データの量がこの制限に達すると、固定されたメモリの合計サイズが制限を超えないように、専用セグメント内のキャッシュ・データが新しいキャッシュ・データのためにフラッシュされることがあります。nbc_pseg_limit が 0 に設定されている場合は、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。</p>
ndd_event_name	<p>目的: ns_alloc/ns_free イベントのトレースが ndd_event_tracing オプションを設定することで使用可能になっている場合に、収集される 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。</p>
ndd_event_tracing	<p>目的: ns_alloc/ns_free トレース・バッファのサイズを指定します。</p> <p>チューニング: このオプションの値がゼロ以外である場合は、すべての ns_alloc/ns_free イベントがカーネル・バッファにトレースされます。0 の値では、このイベント・トレースが使用不可になります。ndd_event_tracing の値が 1024 より大きい場合は、同数の項目がトレース用にカーネル・バッファに割り当てられます。</p>
ndp_mmaxtries	<p>目的: 送信するマルチキャスト NDP (隣接ディスカバリー・プロトコル) パケットの最大数を指定します。</p>
ndp_umaxtries	<p>目的: 送信するユニキャスト NDP (隣接ディスカバリー・プロトコル) パケットの最大数を指定します。</p>
ndpqsize	<p>目的: IP バージョン 6 によって使用される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。</p>
ndpt_down	<p>目的: NDP エントリーをダウンにする時間を 0.5 秒単位で指定します。</p>
ndpt_keep	<p>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。</p>
ndpt_probe	<p>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遅延させる時間を 0.5 秒単位で指定します。</p>
ndpt_reachable	<p>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。</p>
ndpt_retrans	<p>目的: NDP 要求を再送する前に待機する時間を 0.5 秒単位で指定します。</p>

項目 net_buf_size	<p>説明</p> <p>目的: 収集される net_malloc/net_free イベントのバッファ・サイズのリストを指定します。</p> <p>チューニング: net_buf_size の文字列は、サイズのリストを表します。この属性の値が all でない場合は、これらのサイズの net_malloc/net_free イベントのみが収集されます。all の値では、すべてのサイズのイベントが収集されます。</p>
net_buf_type	<p>目的: 収集される net_malloc/net_free イベントのバッファ・タイプのリストを指定します。</p> <p>チューニング: net_buf_type の文字列は、タイプのリストを表します。この文字列が空でなく、all 以外である場合は、そのタイプの net_malloc/net_free イベントのみが収集されます。</p>
net_malloc_frag_mask	<p>目的: これは、各バケットが類似のフラグメントをフルページにプロモーションするように要求するマスクのブール属性として使用されます。</p> <p>チューニング: メモリーの上書き問題をより適切に検出するために、1 より小さいページからフルページに割り当てをプロモーションできるようにします。これは、このようなフラグメントをフルページにプロモーションするように要求する各バケット・サイズのマスクです。メモリーのフラグメントに対してこのオプションを使用可能にすると、パフォーマンスが低下します。</p>
netm_page_promote	<p>目的: ページ・サイズに対してフラグメントのプロモーションを許可するかどうかを指定します。</p> <p>チューニング: このオプションを使用すると、net_malloc_frag_mask で指定されたフラグメント・サイズをページ・サイズにプロモーションすることができます。このオプションを 0 に設定すると、net_malloc_frag_mask に設定されたサイズに関係なくページのプロモーションが使用不可になります。</p>
nonlocsrcroute	<p>目的: 嚴重に送信元で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指定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。</p> <p>チューニング: 値 0 では、外部ホストへのアドレス指定は許可されません。1 の値では、パケットを外部ホストへアドレス指定できます。送信元で厳密に経路指定されなかったパケットは、この属性の影響を受けません。</p>
nstrpush	<p>目的: 単一のストリームにプッシュできるモジュールの最大数を指定します。最小値は 8 です。</p> <p>チューニング: このパラメーターは読み取り専用です。この属性は、オペレーティング・システムのロード時に /etc/pse_tune.conf ファイルで設定できます。</p>
passive_dgd	<p>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。</p> <p>チューニング: 0 の値では passive_dgd が使用不可になり、1 の値では使用中のすべてのゲートウェイに対して使用可能になります。</p>

項目	説明
pmtu_default_age	<p>目的: このオプションは現在使用されていません。これは、UDP アプリケーションがパス MTU の減少を検出できるように IP_DONTFRAG ソケット・オプションを常に設定する必要があるためです。</p> <p>チューニング: ゼロの値では経時は許可されません。デフォルト値は 10 分です。 pmtu_default_age 値は UDP アプリケーションによってオーバーライドできません。pmtu_default_age はランタイム属性です。</p>
pmtu_expire	<p>目的: 参照カウントがゼロのパス MTU エントリーが削除されるまでのデフォルトの時間 (分単位) を指定します。</p> <p>チューニング: 値を 0 にすると、pmtu エントリーは期限切れになりません。</p>
pmtu_rediscover_interval	<p>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 (分単位) を指定します。</p> <p>チューニング: 0 の値ではパス MTU の再ディスカバリーは許可されません。</p>
psebufcalls	<p>目的: ストリームによって割り当てる bufcalls の最大数を指定します。</p> <p>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。これにより、allocb 呼び出しが失敗したときに、ユーザーは bufcall に対する要求を登録できます。システムが再始動するまで、この値を下げることはできません。再始動時に、パラメーターはデフォルト値に戻ります。</p>
psecache	<p>目的: ストリーム・バッファの数を制御します。</p>
psetimers	<p>目的: ストリームによって割り当てるタイマーの最大数を指定します。</p> <p>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タイマー構造を割り当てます。これにより、ストリーム・ドライバーまたはモジュールは timeout 呼び出しを登録できます。システムが再始動するまで、この値を下げることはできません。再始動時に、パラメーターはデフォルト値に戻ります。</p>
rfc1122addrchk	<p>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実行します。</p> <p>チューニング: 0 の値では、アドレス妥当性検査は実行されません。1 の値では、アドレス妥当性検査が実行されます。</p>
rfc1323	<p>目的: RFC 1323 (ハイパフォーマンス用 TCP 拡張) で指定された TCP 拡張を使用可能にします。</p> <p>チューニング: 0 の値では、RFC 拡張がシステム全体のスケールで使用不可になります。1 の値では、すべての TCP 接続が RFC 拡張とのネゴシエーションを試行す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動作をオーバーライドできます。ifconfig コマンドを使用して、各インターフェースごとに rfc1323 ネットワーク・オプションを設定することもできます。</p>
rfc2414	<p>目的: RFC 2414 で記述される TCP の初期ウィンドウの増加を使用可能にします。</p> <p>チューニング: オンの場合、初期ウィンドウは tcp_init_window チューナブルの設定によって決まります。</p>

項目	説明
route_expire	<p>目的: 経路を期限切れにするかどうかを指定します。</p> <p>チューニング:</p> <p>0 の値では、経路を期限切れにすることは許可されません。このオプションに負の値は使用できません。</p>
routevalidate	<p>目的: 経路指定テーブルに新しい経路が追加されるたびに、接続の各キャッシュ経路の妥当性を検査する必要があることを指定します。</p> <p>チューニング:</p> <p>このオプションによって、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を使用できます。0 の値では、キャッシュ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、パフォーマンスがある程度低下することがあります。</p>
rto_high	<p>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・セグメントの再送に使用される最大許容再送数を指定します。</p> <p>チューニング:</p> <p>rto_high は上限係数です。</p>
rto_length	<p>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の値と、TCP データ・セグメントの再送に使用される最大許容再送数を指定します。</p> <p>チューニング:</p> <p>rto_length は時間セグメントの合計数です。</p>
rto_limit	<p>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しきい値と、TCP データ・セグメントの再送に使用される最大許容再送数を指定します。</p> <p>チューニング:</p> <p>rto_limit は rto_low から rto_high の時間セグメントの数です。</p>
rto_low	<p>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值と、TCP データ・セグメントの再送に使用される最大許容再送数を指定します。</p> <p>チューニング:</p> <p>rto_low は下限係数です。</p>
sack	<p>目的: RFC 2018 で記述される TCP 選択肯定応答を使用可能にします。</p> <p>チューニング:</p> <p>1 の値ではすべての TCP 接続で sack のネゴシエーションが行われます。デフォルトはゼロで、ネゴシエーションは使用不可になります。sack 機能にはピア TCP からのサポートが必要です。接続開始時のネゴシエーション・フェーズでそれが決定されます。順序に従わないセグメントを受信する場合、受信側からの選択肯定応答により、受信したデータの送信側に通知され、送信側は欠落しているセグメントのみを再送できるようになります。これにより、不必要に再送されるセグメントが少なくなります。Sack は、データのウィンドウでの複数のパケット・ドロップから迅速に回復するために有効です。</p>
sb_max	<p>目的: TCP および UDP ソケットに許可される最大バッファ・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。</p> <p>チューニング:</p> <p>サイズは、できれば 4096 の倍数になるように増やしてください。最大ソケット・バッファ制限のおよそ 2 倍から 4 倍にする必要があります。</p>

項目	説明
send_file_duration	<p>目的: ネットワーク・バッファ・キャッシュ内でシステム呼び出し send_file がアクセスしたすべてのファイル・オブジェクトに対してキャッシュ妥当性検査の期間を指定します。</p> <p>チューニング: この属性は秒数です。0 の値は、アクセスがあるたびにキャッシュの妥当性が検査されることを意味します。</p>
site6_index	<p>目的: サイトのローカル経路指定のための最大インターフェース数を指定します。</p>
sockthresh	<p>目的: ソケットに割り当てできるネットワーク・メモリ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を予約するために使用します。</p> <p>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の合計量が sockthresh のしきい値に達すると、socket および socketpair システム呼び出しがエラー ENOBUFS で失敗します。着信接続要求はそのまま廃棄されます。既存のソケットは引き続き、さらに多くのメモリを使用できます。sockthresh 属性は thewall 属性のパーセントを表します。</p>
sodebug	<p>目的: 新たに作成されたソケットで SO_DEBUG フラグをオンにするかどうかを指定します。</p>
sodebug_env	<p>目的: 新たに作成されたソケットに対して SODEBUG プロセス環境変数を検査するかどうかを指定します。検査される場合は、これらのソケットで SO_DEBUG フラグがオンになります。</p>
somaxconn	<p>目的: 最大 listen バックログを指定します。</p> <p>チューニング: ビジョ Web サーバー上でこのパラメーターを大きくして、ピーク接続率を処理します。</p>
strctlsz	<p>目的: 単一のシステム呼び出しでストリームに渡してメッセージの制御部分 (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。</p> <p>チューニング: 制御部分がこのサイズを超える putmsg 呼び出しは、ERANGE で失敗します。</p>
strmsgsz	<p>目的: 単一のシステム呼び出し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。</p> <p>チューニング: このサイズを超えるすべての write 呼び出しは複数のメッセージに分割されません。データ部分がこのサイズを超える putmsg 呼び出しは、ERANGE で失敗します。</p>
strthresh	<p>目的: 通常にストリームに割り当てが許可される最大バイト数を指定します。</p> <p>チューニング: strthresh では、このしきい値が渡されると、適切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。このしきい値は出力に適用され、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機能し続けます)。ゼロの値はしきい値がないことを意味します。strthresh 属性は thewall 属性のパーセントを表します。thewall 属性は、net_malloc 呼び出し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。</p>

項目	説明
strturncnt	<p>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処理する要求の最大数を指定します。</p> <p>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、以前にエンキューされた要求を 1 つずつデキューして実行します。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を処理しなければならず、常にその処理でビジーとなり、停止状態になってしまいます。この状態を避けるために、現在実行中のスレッドは strturncnt で指定した数のスレッドのみに対応し、その後別のカーネル・スレッドがすべての保留要求をアクティブにして実行します。</p>
subnetsarelocal	<p>目的: サブネット・マスクと一致するすべてのサブネットを、確立の目的のためにローカルと見なすかどうかを指定します (TCP 最大セグメント・サイズなど)。</p> <p>チューニング: このパラメーターは in_localaddress サブルーチンによって使用されます。デフォルト値の 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。値が 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がローカルです。これは、パフォーマンス結果を伴う構成上の決定です。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が低下する可能性があります。サブネットが同じ MTU を持ち、subnetsarelocal が 0 の場合、TCP セッションは小さい MSS を使用することがあります。</p>
tcp_bad_port_limit	<p>目的: RFC 2481 に記述されているように、Explicit Congestion Notification (明示的輻輳通知) の TCP レベル・サポートを使用可能にします。</p> <p>チューニング: デフォルトはオフ (0) です。オン (1) にすると、すべての接続で ECN 機能とピアとのネゴシエーションが行われます。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。</p>
tcp_cwnd_modified	<p>目的: 特定のソケット・オプションを持つ TCP IP アプリケーションがネットワークの輻輳ウィンドウを調整できるようにします。このパラメーターは、特定の広域ネットワーク (WAN) 環境でしか使用できない場合があります。</p> <p>チューニング: デフォルト値は 0 であり、これはチューニング・パラメーターを使用不可にします。パラメーターを 1 の値にチューニングすると、ネットワーク輻輳ウィンドウを調整することができます。</p>
tcp_ephemeral_high	<p>目的: TCP 一時ポートのために割り当てる最大ポート番号を指定します。</p> <p>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、tcp_ephemeral_low を 1024 に設定します。</p>
tcp_ephemeral_low	<p>目的: TCP 一時ポートのために割り当てる最小ポート番号を指定します。</p> <p>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、tcp_ephemeral_low を 1024 に設定します。</p>

項目	説明
tcp_fastlo	<p>目的: TCP ループバック・トラフィックによって TCP/IP スタック・プロトコル/インターフェース全体をカットオフしてパフォーマンスを向上させることができます。</p> <p>チューニング: 値が 1 の場合、TCP ループバック・トラフィックにより TCP/IP スタック全体をカットオフできます。値を 0 にすると、このオプションが無効になります。</p>
tcp_finwait2	<p>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。</p>
tcp_icmpsecure	<p>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。</p> <p>チューニング: このオプションは、TCP 接続を ICMP アタックから保護するためにオンにしてください。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なっている場合、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 発信元アタックに対して保護されます。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 アタックが大幅に軽減されます。</p>
tcp_init_window	<p>目的: この値は rfc2414 がオンの場合にのみ使用されます (それ以外は無視されます)。</p> <p>チューニング: rfc2414 がオンで、この値が 0 の場合、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。 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できます。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、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上がります。</p>
tcp_inpcb_hashtab_siz	<p>目的: TCP 接続のための inpcb ハッシュテーブルのサイズを指定します。</p> <p>チューニング: 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実装されます。大きいテーブルは、リンクされたハッシュ・チェーンが小さく、平均トラバースル時間を下回るが、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。この値は素数でなければなりません。このオプションはパフォーマンスに影響するので、使用には厳重な注意が必要です。値の変更が必要と思われる場合には、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する可能性があります。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。</p>
tcp_keepcnt	<p>目的: tcp_keepcnt は、接続の終了前に送信できたキープアライブ・プローブの数を表します。</p>
tcp_keepidle	<p>目的: 接続をアクティブに保持する時間の長さを 0.5 秒単位で指定します。</p>
tcp_keepinit	<p>目的: TCP 接続のための初期タイムアウト値を 0.5 秒単位で設定します。</p>

項目 tcp_keepintvl	<p>説明</p> <p>目的: 接続の妥当性を検査するために送信されるパケット間の間隔を 0.5 秒単位で指定します。</p> <p>チューニング: 例えば、0.5 秒を 150 個にすると、妥当性検査プローブの間隔が 75 秒になります。これにより、TCP は接続がまだ有効であることを認識し、接続がアイドル状態である場合はその接続をオープンのまま保持することができます。これは、最低限のパフォーマンス結果を伴う構成決定です。変更しないことをお勧めします。この間隔を大幅に短縮した場合、処理コストおよび帯域幅コストが大きくなります。</p>
tcp_limited_transmit	<p>目的: RFC 3042 で記述される TCP の消失リカバリーを拡張する機能を使用可能にします。</p> <p>チューニング: 値 1 の場合はこのオプションが使用可能になり、ゼロの場合はこのオプションが使用不可になります。</p>
tcp_low_rto	<p>目的: パケット・ドロップが発生した接続の TCP 再送タイムアウト (RTO) をミリ秒単位で指定します。</p> <p>チューニング: 1 ティックは 10 ミリ秒 (100 分の 1 秒) です。オプション <code>timer_wheel_tick</code> は、<code>tcp_low_rto</code> オプションを設定する前にゼロ以外の値に設定しておく必要があります。また、<code>tcp_low_rto</code> はゼロまたは <code>timer_wheel_tick</code> 値の 10 の倍数にすることができます。このチューナブルを使用すると、TCP はパケットのタイムアウトに小さいタイムアウト値を使用でき、高速ネットワークで再送できます。通常の TCP 再送タイムアウトは 1.5 秒です。</p>
tcp_maxburst	<p>目的: バックツーパーク・パケットをその宛先に転送するために一時停止する前に TCP が送信できるバックツーパーク・パケットの数を指定します。</p> <p>チューニング: これはルーターが TCP パケットの大きいバーストを処理できず、その一部をドロップしている場合に有効です。0 の値は、一時停止前のバックツーパーク・パケット数には制限がないことを意味します。</p>
tcp_mssdflt	<p>目的: リモート・ネットワークとの通信で使用されるデフォルトの最大セグメント・サイズ。</p> <p>チューニング: tcp_mssdflt は、パス MTU ディスカバリーが使用不可である場合、またはパス MTU ディスカバリーがパス MTU をディスカバリーできない場合のみ使用されます。各インターフェースごとに tcp_mssdflt ネットワーク・オプションを設定することもできます (ISNO オプションについては資料を参照してください)。データを (MTU - 40) バイトに制限すると、可能な場合は、いっぱいであるパケットのみが送信されます。</p>
tcp_nagle_limit	<p>目的: これはバイト単位の Nagle アルゴリズムのしきい値で、Nagle を使用不可にするために使用できます。</p> <p>チューニング: デフォルトでは、Nagle がオンです。Nagle を使用不可にするには、この値を 0 または 1 に設定します。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。</p>

項目	説明
tcp_nagleoverride	<p>目的: tcp_nagle_limit オプションを設定すると Nagle アルゴリズムがシステム全体でオフになり、ソケットに tcp_nodelay オプションを設定するとその固有の接続に対して Nagle アルゴリズムがオフになります。それに対し、tcp_nagleoverride を設定すると接続中に特定の状態に対してのみ Nagle アルゴリズムが使用不可になります。</p> <p>チューニング: 値が 1 の場合は、接続内の特定の TCP パケットに対してのみ Nagle アルゴリズムが使用不可になります。</p>
tcp_ndebug	<p>目的: tcp_debug 構造の数を指定します。</p>
tcp_newreno	<p>目的: RFC 2582 で記述される TCP の高速リカバリー・アルゴリズムの変更を使用可能にします。</p> <p>チューニング: これは TCP の高速再送アルゴリズムの制限を修正し、ウィンドウ内の複数のパケットがドロップされた場合に、ドロップされたパケットから高速にリカバリーします。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 修正は送信側のみで行われます。</p>
tcp_nodelayack	<p>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 が使用不可の場合、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。</p> <p>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実装でのバグを解決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、ネットワーク転送のパフォーマンスが著しく向上する可能性があります。</p>
tcp_pmtu_discover	<p>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。</p> <p>チューニング: 値が 0 の場合は TCP アプリケーションのパス MTU ディスカバリーが使用不可になり、値が 1 の場合は使用可能になります。</p>
tcp_recvspace	<p>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 により使用されるウィンドウ・サイズに影響します。</p> <p>チューニング: 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。各インターフェースごとに tcp_recvspace ネットワーク・オプションを設定することもできます (インターフェース固有のネットワーク・オプション (ISNO) については資料を参照してください)。現在、ほとんどのインターフェースで、このチューナブルが ISNO のデフォルトに設定されています。tcp_recvspace 属性には、sb_max 属性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。</p>

項目	説明
tcp_sendspace	<p>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。</p> <p>チューニング: 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 (最適なウィンドウ=帯域幅 * 平均往復時間)。各インターフェースごとに tcp_sendspace ネットワーク・オプションを設定することもできます (インターフェース固有のネットワーク・オプション (ISNO) については資料を参照してください)。現在、ほとんどのインターフェースで、このチューナブルが ISNO のデフォルトに設定されています。tcp_sendspace 属性には、sb_max 属性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。</p>
tcp_tcpsecure	<p>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。</p> <p>チューニング: このオプションは、次の 3 つのぜい弱性のうち 1 つ以上から TCP 接続を保護する場合に使用します。最初のぜい弱性は、接続を異常終了させるための、確立された接続に対する偽の SYN の送信に関するものです。tcp_tcpsecure の値 1 は、このぜい弱性からの保護を提供します。2 番目のぜい弱性は、接続を異常終了させるための、確立された接続に対する偽の RST の送信に関するものです。tcp_tcpsecure の値 2 は、このぜい弱性からの保護を提供します。3 番目のぜい弱性は、確立された TCP 接続における偽データの流入に関するものです。tcp_tcpsecure の値 4 は、このぜい弱性からの保護を提供します。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、これら 3 つのぜい弱性の組み合わせから接続を保護します。</p>
tcp_timewait	<p>目的: tcp_timewait オプションは、timewait 状態に接続を保持する時間を構成するために使用します。</p> <p>チューニング: 15 秒間隔で指定できます。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低下します。</p>
tcp_ttl	<p>目的: TCP パケットの存続時間をティック単位で指定します。</p> <p>チューニング: 1 ティックは 0.6 秒 (100 ティックが 1 分) です。</p>
tcprexmtthresh	<p>目的: TCP を高速再送フェーズに移動させるための、連続した重複肯定応答の数を指定します。</p> <p>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一方で、ネットワークは混み合っていない場合は、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。</p>
tcptr_enable	<p>目的: tcptr コマンドを使用して作成されたポリシーで定義される TCP トラフィック規定を有効にします。値 0 は無効であることを示します。ゼロ以外の値は、トラフィック規定が有効であることを示します。</p> <p>チューニング: 値を 0 にすると、このオプションが無効になります。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があるサーバーではオンにする必要があります。</p>

項目	説明
thewall	<p>目的: メモリー・プールに割り当てられるメモリーの最大量 (キロバイト数) を指定します。</p> <p>チューニング: 現在は設定できません。</p>
timer_wheel_tick	<p>目的: タイマー・ホイールのスロット間隔をティック単位で指定します。1 ティックは 1000/HZ=10ms です。</p> <p>チューニング: この属性は、tcp_low_rto 属性と使用して、TCP タイムアウト値を小さい単位に減らします。</p>
tn_filter	<p>目的: このオプションは、Trusted AIX 環境に対してのみ有効です。このオプションがこの環境で使用不可である場合は、MAC 検査が IP 層でバイパスされます。</p>
udp_bad_port_limit	<p>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信したら、UDP はこのようなパケットに回答して 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。</p> <p>チューニング: 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場合に UDP が ICMP エラーの送信を停止することを示す数です。</p>
udp_ephemeral_high	<p>目的: UDP 一時ポートのために割り当てる最大ポート番号を指定します。</p>
udp_ephemeral_low	<p>目的: UDP 一時ポートのために割り当てる最小ポート番号を指定します。</p>
udp_inpcb_hashtab_siz	<p>目的: UDP 接続のための 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実装されます。大きいテーブルは、リンクされたハッシュ・チェーンが小さく、平均トラバーサル時間を下回るが、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。</p> <p>チューニング: この値は素数でなければなりません。このオプションはパフォーマンスに影響を及ぼすため、使用する際には特に注意が必要です。値の変更が必要と思われる場合には、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する可能性があります。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。</p>
udp_pmtu_discover	<p>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。</p> <p>チューニング: パス MTU ディスカバリーを使用するために、UDP アプリケーションを書く必要があります。0 の値はこの機能を使用不可にするのに対して、1 の値はそれを使用可能にします。</p>
udp_recvspace	<p>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。</p> <p>チューニング: udp の netstat -s レポートで n がゼロでない場合に変更します。n ソケット・バッファがオーバーフローしています。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。サイズは、できれば 4096 の倍数になるように増やしてください。</p>

項目	説明
udp_send_perf	<p>目的 ネットワークを介してパケットを送信するのに使用されるアドレス情報とメモリー・バッファ (mbufs) をキャッシュすることで UDP の送信パフォーマンスを向上させます。</p> <p>チューニング: デフォルト値は 0 であり、キャッシングを使用不可に設定します。キャッシングを使用可能にするには、値 1 を指定してください。例えば、キャッシングを使用可能にするために、以下のコマンドを入力します。</p> <pre>no -o udp_send_perf=1</pre>
udp_sendspace	<p>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・サイズ (バイト数) を指定します。</p> <p>チューニング: udp_sendspace 属性には、sb_max 属性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。udp_sendspace は、少なくともアプリケーションが送信する最大データグラム・サイズと同じ大きさでなければなりません。サイズは、できれば 4096 の倍数になるように増やしてください。</p>
udp_ttl	<p>目的: UDP パケットの存続時間 (秒単位) を指定します。</p>
udpcksum	<p>目的: UDP チェックサムをオン/オフにすることを可能にします。</p> <p>チューニング: 0 の値ではこれがオフになるのに対して、1 の値ではこれがオンになります。</p>
use_sndbufpool	<p>目的: mbuf クラスターのキャッシュを使用可能にして、パフォーマンスを向上させます。</p> <p>チューニング: この値が使用不可の場合、1 つの mbuf クラスターを割り当てるために、AIX は、1 つのクラスター・バッファと、それを指す mbuf バッファを割り当てる必要があります。したがって、2 つのバッファ割り当て操作が必要になります。同様に、クラスターを解放する場合には、2 つのバッファ解放操作が必要になります。このオプションを使用可能にすると、AIX はクラスターのキャッシュを、使用中の各クラスター・サイズごとに保持します。これにより、MBUF クラスターの割り当てと解放のためにオーバーヘッドを削減することでパフォーマンスが向上します。デフォルト値の 1 の場合、このオプションがシステム全体で使用可能になります。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できます。</p>

互換モード

pre520tune 属性 **sys0** で制御される、5.2 より前の互換モードで実行する場合、**AIX 5.2** 互換モードを参照してください。パラメーターのリポート値は、タイプ **Bosboot** の場合を除き、適用されません。5.2 より前の互換モードでは、ブート時にこれらの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、引き続き、ブート・シーケンス中に呼び出されるスクリプトにチューニング・コマンドの呼び出しを埋め込むことによって行われます。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、「*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。

セキュリティ

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権命令を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権についての詳細情報は、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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`
- システムをインターネット・プロトコル・ネットワークでのインターネットワーク・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`
- 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`
- `udp_ephemeral_high` オプションに関する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`
- `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`
- 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`
- 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`
- 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`
- 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`
- 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`
- 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

通信およびネットワーク

AIX 5.2 互換モード

インターネット・プロトコル

カーネル・チューニング

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

項目	説明
-p pid	<i>pid</i> 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。

項目	説明
126	<i>Command</i> 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 <i>Command</i> 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
```

関連資料:

86 ページの『**nice** コマンド』

関連情報:

cs コマンド

sh コマンド

signal コマンド

enotifyevent コマンド、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 にメールで送信します。

構文

enotifyevent [-h] [*user-ID*]

notifyevent [-h] [*user-ID*]

説明

enotifyevent スクリプトは常にメッセージを英語で戻します。 **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。

これらのスクリプトは、イベント応答リソース・マネージャー (ERRM) が環境変数内にポストしたイベント情報をキャプチャーします。この環境変数は、イベントの発生時に ERRM が生成します。これらのスクリプトは、イベント応答リソースが実行す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も使用できます。

ERRM 環境変数に関するイベント情報が戻されます。また、イベント情報には以下も含まれます。

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。

AIX では、これらのスクリプトは **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。

AIX では、 *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。

その他のプラットフォームの場合には、`log_file` のサイズは制限されず、それ自身を上書きしません。ファイル・サイズは、管理者が定期的にエントリーを除去しないかぎり、無限に増え続けます。

`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` フラグを指定すると、スクリプトの使用方法に関するステートメントが標準出力に書き込まれます。

例

1. 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 2002

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

`/opt/rsct/bin/enotifyevent`

`enotifyevent` スクリプトが入っています。

`/opt/rsct/bin/notifyevent`

`notifyevent` スクリプトが入っています。

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。ファイルが指定されていないか、最後のパラメーターとして - (マイナス符号) フラグが指定されている場合には、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

項目	説明
-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	水平スペース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。
-nNumber	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、 <i>List</i> 変数で指定されたページのみを出力します。 <ul style="list-style-type: none">• 範囲 <i>Start-Stop</i> は、ページ <i>Start</i> から <i>Stop</i> 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 <i>-Stop</i> は、先頭ページからページ <i>Stop</i> までの出力を意味します。• 最後の <i>Start-</i> は、ページ <i>Start</i> 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、指定したページが印刷されます。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 <i>List</i> 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。このブロークン・パイプ・メッセージは何の問題も示しておらず、無視することができます。
-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 <i>A</i> を指定値に設定します。 <i>A</i> 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。

項目
-TName

説明
指定された印刷装置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和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す。

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 で 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。

項目	説明
-T <i>Name</i> (続き)	<p>2631 正規モードの Hewlett-Packard 2631 プリンター。</p> <p>2631-c 圧縮モードの Hewlett-Packard 2631 プリンター。</p> <p>2631-e 拡張モードの Hewlett-Packard 2631 プリンター。</p> <p>300 DASI-300 プリンター。</p> <p>300-12 1 インチ当たり 12 文字に設定している DASI-300 端末。</p> <p>382 DTC-382。</p> <p>4000a Trendata 4000a 端末 (4000A)。</p> <p>450 DASI-450 (Diablo Hyterm) プリンター。</p> <p>450-12 1 インチ当たり 12 文字に設定している DASI-450 端末。</p> <p>832 Anderson Jacobson 832 端末。</p> <p>8510 C.ITOH プリンター。</p> <p>tn300 GE Terminet 300 端末。</p> <p>X TX 印刷トレイン搭載のプリンター。</p> <p>300s DASI-300s プリンター (300S)。</p> <p>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)。 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。<i>Number</i> 変数を指定しない場合は、0 になります。</p>
-u <i>Number</i>	.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。
-z	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』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

項目	説明
/usr/share/lib/tmac/tmac.*	標準マクロ・ファイルを指すポインタが入っています。
/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。

関連資料:

24 ページの『**neqn** コマンド』

関連情報:

col コマンド

mm コマンド

nroff および **troff** 入力

nroff および **troff** コマンド用 **nroff** および **troff** 要求

nslookup コマンド

目的

インターネット・ドメインのネーム・サーバーを対話式に照会します。

構文

```
nslookup [ - option ] [ name | - ] [ server ]
```

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。**nslookup** コマンドが非対話式モードに入るのは、検索対象のホスト名またはホストのインターネット・アドレスを最初の引数として指定した場合です。2 番目のオプションの引数には、ネーム・サーバーのホスト名またはアドレスを指定します。コマンド・ラインに各オプションを指定可能なのは、オプションが各引数の前にあり、かつ、ハイフンが接頭部として付けられている場合です。例えば、デフォルトの照会タイプをホスト情報に変更し、初期タイムアウトを 10 秒に変更するには、下記のコマンドを入力します。

```
nslookup -query=hinfo -timeout=10
```

対話式コマンド

項目	説明
host [server]	現行のデフォルト・サーバーを使用、または指定されている場合はサーバーを使用するホスト用の情報を検索します。そのホストがインターネット・アドレスであり、照会タイプが A または PTR の場合は、 nslookup コマンドはそのホストの名前を戻します。そのホストが名前であり、かつ、末尾にピリオドが付いていない場合、検索リストを使用してその名前を限定します。現行ドメイン内に存在しないホストを探す場合、その名前の後ろにピリオドを 1 つ付けます。
server Domain lserver Domain	デフォルトのサーバーを <i>Domain</i> 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。
exit	プログラムを終了します。

項目

set *Keyword*[=*Value*]

説明

検索に影響を及ぼす状態情報を変更します。以下のいずれかのキーワードを指定できます。

all set を行うために頻繁に使用されるオプションの現在値を表示します。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値に変更します。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。

CH カオス・クラス。

HESIOD

Hesiod クラス。

ANY ワイルドカード (上記のいずれか)。

[no]debug

デバッグ・モードをオンにします。デフォルト値は **nodebug** 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** 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。

[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=*valuetype*=*value*

value パラメーターで指定したタイプに、情報照会のタイプを変更します。デフォルト値は **A** です。

[no]recurse

その情報がない場合、他のサーバーを照会する旨をネーム・サーバーに通知します。デフォルトは **recurse** です。

retry=*number*

number パラメーターで指定した数値に再試行回数を設定します。

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。

[no]vc サーバーへの要求送信時に、常時、仮想回線を使用します。デフォルト値は **novc** です。

[no]fail ネーム・サーバーが、そのような応答時に **SERVFAIL** または委託 (**nofail**) または照会終了 (**fail**) で応答する場合、次のネーム・サーバーを試行します。デフォルトは **nofail** です。

ファイル

項目	説明
<code>/etc/resolv.conf</code>	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。

関連資料:

- 2 ページの『`named-checkconf` コマンド』
- 8 ページの『`named9` デーモン』
- 872 ページの『`rndc-confgen` コマンド』

関連情報:

- `host9` コマンド
- `dnssec-keygen` コマンド
- `dnssec-signzone` コマンド

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 7.1 は BIND バージョン 9 のみをサポートしています。BIND 8 アプリケーション・コードは、現在 AIX 7.1、および `named` デーモンの `named9` へのリンクと `nsupdate` の `nsupdate4` へのリンクから削除されています。別のバージョンの `nsupdate` を使用する場合は、`nsupdate` コマンドに応じてシンボリック・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` (現在は AIX 7.1 から削除されています) と一緒に使用できますが、`nsupdate9` はセキュリティ処理が異なるため `named9` と一緒に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。

関連資料:

『nsupdate4 コマンド』

関連情報:

bootp コマンド

ネーム・サーバー・レゾリューション

ドメイン・ネーム・レゾリューションの計画

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)

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。

項目	説明
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すが、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

項目	説明
-a	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。
-d <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。
-g	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す。
-h <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。
-i	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。
-k <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。
-p <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。
-q	出力をオフにします。
-r <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。
-s " <i>CommandString</i> "	スペースまたはコロンの区切られた内部コマンドのセットです。
-v	詳細出力をオンにします。
-?	コマンド・ライン・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。

項目	説明
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 (";" の箇所を改行キーを押します)。

```
r;d;a;*;a;a;9.3.145.2;s;300;3110400;x;q
```

エンタリーに 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・ライン・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

項目	説明
<code>/usr/sbin/nsupdate4</code>	<code>nsupdate4</code> コマンドが入っています。
<code>/usr/sbin/named</code>	DNS サーバーが入っています。

関連情報:

DHCP クライアント構成ファイル

DHCP サーバー構成ファイル

bootp 構成ファイル

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)

TCP/IP デーモン

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・ライン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

項目	説明
<code>-d</code>	<code>nsupdate8</code> 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。
<code>-v</code>	<code>nsupdate8</code> 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、各パケットはブランク行 (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。

項目 <i>section</i>	<p>説明 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。</p> <p>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。</p>
<i>opcode</i>	<p>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。 このレコードを使用して行う操作を定義します。</p> <p>値は次のとおりです。 前提条件操作:</p> <p>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。 <i>ttl</i> 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。 <i>T_ANY</i> 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。</p> <p>nydomain その名前が存在しているか確認することを示します。<i>ttl</i> 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 <i>T_NONE</i> 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。</p> <p>nxxrset 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 <i>ttl</i> は、検索を制限することができます。タイプは必須です。</p> <p>nyxrset 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 <i>ttl</i> 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。</p> <p>値は次のとおりです。 更新操作:</p> <p>追加 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。<i>ttl</i> 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。</p> <p>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 <i>T_ANY</i> をとります。 <i>ttl</i> およびクラスはオプションです。 <i>ttl</i> は、指定されると 0 にリセットされます。</p>
<i>name</i>	テストまたは修正が行われている DNS エントリーの名前。
<i>[ttl]</i>	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。
<i>[class]</i>	ゾーンに追加されるレコードのクラス。値は <i>IN</i> 、 <i>HESIOD</i> 、および <i>CHAOS</i> です。メッセージ全部のデフォルトは <i>IN</i> です。
<i>[type]</i>	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、 <i>A</i> 、 <i>NS</i> 、 <i>CNAME</i> 、 <i>SOA</i> 、 <i>MB</i> 、 <i>MR</i> 、 <i>NULL</i> 、 <i>WKS</i> 、 <i>PTR</i> 、 <i>HINFO</i> 、 <i>MINFO</i> 、 <i>MX</i> 、 <i>TXT</i> 、 <i>RP</i> 、 <i>AFSDB</i> 、 <i>X25</i> 、 <i>ISDN</i> 、 <i>RT</i> 、 <i>NSAP</i> 、 <i>NSAP_PTR</i> 、 <i>PX</i> 、および <i>LOC</i> です。注: <i>CNAME</i> タイプは、 <i>TSIG</i> および <i>TKEY</i> 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 <i>BIND 8</i> ではサポートされていません。
<i>[data]</i>	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 <i>DNS</i> サーバー・ゾーン・ファイルの <i>DOMAIN</i> 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。

関連資料:

236 ページの『nsupdate コマンド』

1 ページの『named デーモン』

関連情報:

DOMAIN Cache

TCP/IP デーモン

ネーム・サーバー・レゾリューション

nsupdate9 コマンド

目的

動的 DNS 更新ユーティリティ。

構文

```
nsupdate9 [-d] [-y [ hmac: ] keyname: secret | -k keyfile] [ -t timeout ] [ -u udptimeout ] [ -r udpretries ] [-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、あるいは、RFC3535 と RFC2931 に記載された SIG(0) レコードを使用します。署名は、**nsupdate9** とネーム・サーバーだけが知っている共有秘密鍵に依存しています。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。SIG(0) は公開鍵暗号方式を使用します。SIG(0) 鍵を使用するには、ネーム・サーバーがサービスするゾーン内の KEY レコードに、その公開鍵を保管する必要があります。**nsupdate9 does not read /etc/named.conf.**

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鍵を提供します。デフォルト・タイプは HMAC-MD5 です。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**) から共有秘密鍵を読み取ります。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、`[hmac:] keyname:secret` から署名が生成されます。ここで、*keyname* は鍵の名前であり、*secret* は base64 でエンコードされた共有秘密鍵です。**-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・ライン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。

また、**-k** フラグを使用して、動的 DNS 更新要求の認証に使用する SIG(0) 鍵を指定します。この場合、指定された鍵は HMAC-MD5 鍵ではありません。

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。ただし、その要求が、TCP 使用時のケースで UDP 要求に適合するには大きすぎる場合はその限りではありません。**-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

項目	説明
-d	nsupdate9 をデバッグ・モードで操作する。
-k <i>keyfile</i>	ファイル <i>keyfile</i> から共有秘密鍵を読み取る。
-r <i>udpretries</i>	UDP 再試行回数を設定します。デフォルトは 3 です。ゼロの場合は更新要求を 1 回だけ再試行します。
-t <i>timeout</i>	更新要求を異常終了させる前に待たせる最大時間を設定します。デフォルト値は 300 秒です。タイムアウトを使用不可にするには、ゼロを使用することができます。
-u <i>udptimeout</i>	UDP 再試行間隔を設定します。デフォルトは 3 秒です。ゼロにすると、この間隔は UDP 再試行のタイムアウト間隔と回数から計算されます。
-v	nsupdate9 が TCP 接続を使用するようにする。
-y <code>[hmac:] keyname:secret</code>	<code>keyname:secret</code> から署名を生成する。

パラメーター

項目	説明
<i>filename</i>	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、更新は拒否されます。

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リース・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。

項目	説明
server [servername] [port]	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。 server ステートメントを指定しないと、 nsupdate9 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。
local [address] [port]	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。 local ステートメントを指定しないと、 nsupdate9 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。
zone [zonename]	すべての更新をゾーン <i>zonename</i> に行うことを指定します。 zone ステートメントを指定しないと、 nsupdate9 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。
key [name] [secret]	すべての更新が <i>keyname</i> <i>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。 key コマンドは、 -y または -k を使用して、コマンド・ラインに指定されたすべてのキーをオーバーライドします。
prereq nxdomain [domain-name]	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。
prereq yxdomain [domain-name]	<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)。
prereq nxrrset [domain-name] [class] [type]	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。
prereq yxrrset [domain-name] [class] [type]	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。
prereq yxrrset [domain-name] [class] [type] [data...]	共通の <i>type</i> 、 <i>class</i> 、および <i>domain-name</i> を共有する、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。
update delete [domain-name] [ttl] [class] [type] [data...]	<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。
update add [domain-name] [ttl] [class] [type] [data...]	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。
show	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。
send	現行メッセージを送信します。これはブランク行の入力と同等です。
answer	その応答を表示します。

セミコロンで始まる行はコメントであり、無視されます。

セキュリティー

RBAC ユーザーおよび **Trusted AIX ユーザー**への注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティー」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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することはありません。2 つの更新の間に空白行または **send** コマンドを挿入して、個別に更新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
> send
```

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 レコードがすべて削除され、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) を持ちます。

ファイル

項目	説明
/etc/resolv.conf	デフォルト・ネーム・サーバーを識別します。
K{name}.+157.+(random).key	dnssec-keygen(8) により作成される HMAC-MD5 キーの Base-64 エンコード。
K{name}.+157.+(random).private	dnssec-keygen(8) により作成される HMAC-MD5 キーの Base-64 エンコード。

関連資料:

- 2 ページの『named-checkconf コマンド』
- 8 ページの『named9 デーモン』
- 871 ページの『rndc コマンド』
- 872 ページの『rndc-confgen コマンド』

関連情報:

- dnssec-keygen コマンド
- dnssec-signzone コマンド

ntpd4 デーモン

目的

Network Time Protocol (NTP) デーモン。

構文

```
ntpd4 [ -4 ] [ -6 ] [ -a ] [ -A ] [ -b ] [ -c conffile ] [ -d ] [ -D level ] [-f driftfile] [ -g ] [-i jaildir] [ -k keyfile] [-l logfile] [-L] [ -n ] [ -N ] [ -p pidfile] [ -P priority ] [-q] [-r broadcastdelay] [ -s statsdir] [-t key] [ -u user[:group]] [ -U interface update interval] [ -v variable] [-V variable] [-x]
```

説明

ntpd プログラムは、インターネット標準時間サーバーと同期してシステム時刻を設定して維持するオペレーティング・システム・デーモンです。**ntpd** プログラムは、Network Time Protocol (NTP) バージョン 4 の完全な実装となっており、RFC-1305 で定義されているようにバージョン 3 との互換性もあります。また、RFC-1059 と RFC-1119 でそれぞれ定義されているようにバージョン 1 と 2 との互換性もあります。通常、**ntpd** プログラムは 64 ビットの浮動小数点数演算モードで計算します。**ntpd** は、232 ピコ秒の精度を保つ必要がある場合は 64 ビットの固定小数点モードで計算します。既存のワークステーションやネットワークではこの究極の 232 ピコ秒の精度は達成できませんが、この精度は将来のギガヘルツ級の CPU クロックやギガビット LAN において必要となる可能性があります。

周波数の規則

開始時の **ntpd** の動作は、周波数ファイル (通常は **ntp.drift**) によって異なります。このファイルには、クロック周波数の誤差の最新の予測が含まれています。**ntpd** デーモンが開始したときにこのファイルが存在しない場合、**ntpd** は特定のシステム・クロック発振器の時刻誤差と周波数誤差に素早く適合するために設計された特殊モードに入ります。これには約 15 分を要します。その後、時刻と周波数は公称値に設定され、**ntpd** は通常の動作モードに入ります。このモードでは、時刻と周波数が連続的にサーバーに対して追跡されます。1 時間後、周波数ファイルが作成され、現在の周波数オフセットがこのファイルに書き込まれます。**ntpd** が開始したときにこのファイルが存在する場合、**ntpd** の周波数はこのファイルから初期化され、**ntpd** は通常の動作モードに入ります。その後 1 時間の間隔で、現行の周波数オフセットがこのファイルに書き込まれます。

動作モード

ntpd プログラムは、対称アクティブ/パッシブ、クライアント/サーバー、およびブロードキャスト/マルチキャストを含む複数のモードのいずれかで動作します。通常、**ntpd** は連続的に動作し、その一方で小さい周波数の変化をモニターし、クロックを究極の精度に向けて調整します。**ntpd** はワнтаイム・モードで動作することもあります。このモードでは、時刻は外部サーバーから設定され、周波数は以前に記録された周波数ファイルから設定されます。ブロードキャストまたはマルチキャスト・クライアントは、リモート・サーバーをディスカバーし、サーバー/クライアント間の伝搬遅延修正係数を計算し、それ自体を自動的に構成することができます。これにより、ローカル環境に固有の構成を詳細に指定せずに、ワークステーション群を配置することができます。

デフォルトでは、**ntpd** は連続モードで実行されます。このモードでは、複雑な状態マシンにより決定される間隔で、いくつかの外部サーバーがそれぞれポーリングされます。状態マシンは、付随して起こる往復遅延時間のジッターと発振器周波数の逸脱を測定し、ヒューリスティックなアルゴリズムを使用して最適なポーリング間隔を決定します。通常、ほとんどの稼働環境で、この状態マシンは 64 秒の間隔で開始し、最終的に 1024 秒まで段階的に増やします。サーバーのアクセス集中を避けるために、若干のランダム変動が導入されます。加えて、しばらくの間サーバーに到達できない場合は、ネットワークのオーバーヘッドを減らすために、ポーリング間隔が 1024 秒まで段階的に増えます。

ntpd が連続的に実行されることが実用的でない場合があります。一般的な回避策は、**ntpdate** プログラムを **cron** ジョブから指定の時刻に実行することでした。しかし、このプログラムには、きめ細かい **ntpd** の信号処理、誤差検査、および緩和アルゴリズムはありません。**-q** オプションはこの目的のためのものです。このオプションを設定すると、**ntpd** はクロックを初めて設定した直後に終了します。クロックの初期設定の手順は、連続モードのものと同じです。ほとんどのアプリケーションでは、**server** 構成コマンドと一緒に **iburst** コマンドを指定します。このコマンドにより、メッセージが一斉に連続して交換され、データが整えられ、クロックが約 10 秒で設定されます。応答が受信されない場合は、数分後にデーモンがタイムアウトして終了します。ある一定の期間の後、応答が受信されない場合は、**ntpdate** プログラムが停止します。

フラグ

項目	説明
-4	IP バージョン 4 ネーム・スペースに対するホスト名の DNS 解決を強制実行します。
-6	IP バージョン 6 ネーム・スペースに対するホスト名の DNS 解決を強制実行します。
-a	ブロードキャスト・クライアント、マルチキャスト・クライアント、および対称パッシブ・アソシエーションの暗号認証を要求します。これはデフォルト値です。
-A	ブロードキャスト・クライアント、マルチキャスト・クライアント、および対称パッシブ・アソシエーションの暗号認証を要求しません。
-b	クライアントがブロードキャスト・サーバーと同期できるようにします。
-c conffile	構成ファイルの名前とパスを指定します。デフォルトは <code>/etc/ntp.conf</code> です。
-d	デバッグ・モードを指定します。このオプションは複数回指定できますが、指定するごとに表示がより詳細になります。
-D level	デバッグ・レベルを直接指定します。
-f driftfile	周波数ファイルの名前とパスを指定します。デフォルトは <code>/etc/ntp.drift</code> です。これは driftfile driftfile 構成コマンドと同じ動作です。
-g	制限なしに時刻を任意の値に設定できるようにします。これは一度だけ指定できます。 ntpd コマンドは、オフセットがパニックしきい値 (デフォルトは 1000 秒) を超えた場合に、システム・ログにメッセージを出力して終了します。しきい値がその後を超えた場合、 ntpd はシステム・ログにメッセージを出力して終了します。このオプションは、 -q オプションおよび -x オプションと一緒に使用できます。
-i jaildir	chroot コマンドにより、サーバーが jaildir ディレクトリに誘導されます。また、このオプションは、サーバーが始動時に root 特権の除去を試行することを意味します (それ以外の場合、 chroot は極めて低いセキュリティを提供します)。これは、オペレーティング・システムで完全な root 特権を持たずにサーバーを実行できる場合のみ使用できます。 -u オプションを指定する必要があります。

項目	説明
-k <i>keyfile</i>	対称鍵ファイルの名前とパスを指定します。デフォルトは <code>/etc/ntp.keys</code> です。これは <code>keys keyfile</code> 構成コマンドと同じ動作です。
-l <i>logfile</i>	ログ・ファイルの名前とパスを指定します。デフォルトは、システム・ログ・ファイルです。これは <code>logfile</code> 構成コマンドと同じ動作です。
-L	仮想 IP を <code>listen</code> しません。デフォルトでは、 <code>listen</code> します。
-n	フォークしません。
-N	オペレーティング・システムで許可される範囲内の、最も高い優先順位で <code>ntpd</code> を実行します。
-p <i>pidfile</i>	<code>ntpd</code> プロセス ID の記録に使用するファイルの名前とパスを指定します。これは <code>pidfile pidfile</code> 構成コマンドと同じ動作です。
-P <i>priority</i>	オペレーティング・システムで許可される範囲内の、指定された優先順位で <code>ntpd</code> を実行します。
-q	最初にクロックを設定した直後に <code>ntpd</code> を終了します。この動作は、廃止予定の <code>ntpddate</code> プログラムの動作に類似しています。このオプションは、 -g オプションおよび -x オプションと一緒に使用できます。 注: カーネル時刻規則は、このオプションでは無効となります。
-r <i>broadcastdelay</i>	ブロードキャスト/マルチキャスト・サーバーからクライアントまでのデフォルトの伝搬遅延を指定します。このオプションは、プロトコルで遅延を自動的に計算できない場合のみ必要となります。
-s <i>statsdir</i>	統計機能によって作成されるファイルのディレクトリ・パスを指定します。これは <code>statsdir</code> 構成コマンドと同じ動作です。
-t <i>key</i>	キー番号をトラステッド鍵リストに追加します。このオプションは複数回指定できます。
-u <i>user[:group]</i>	切り替えるユーザーを指定します。オプションでグループも指定できます。このオプションは、オペレーティング・システムで完全な <code>root</code> 特権を持たずにサーバーを実行できる場合のみ使用できます。
-U <i>interface update interval</i>	新規および削除済みネットワーク・インターフェースを選択するためにインターフェース・リストをスキャンする間隔を秒数で指定します。動的なインターフェース・リストの更新を使用不可にするには、0 に設定します。デフォルトでは、5 分ごとにスキャンします。
-v <i>variable</i>	デフォルトでリストされるシステム変数を追加します。
-V <i>variable</i>	
-x	オフセットが <code>step</code> しきい値 (デフォルトは 128 ミリ秒) より小さい場合は時刻を微調整により修正し、このしきい値を超える場合はすぐに時刻を修正します。このオプションは、手動でクロックを設定するための精度ウィンドウの範囲内である 600 秒にしきい値を設定します。

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

シンボリック・リンク `/usr/sbin/xntpd` は、デフォルトで、NTP v3 デーモン (`/usr/sbin/ntp3/xntpd`) を指します。NTP v4 デーモン (`/usr/sbin/ntp4/ntpd4`) を実行するには、v4 デーモンを指すようにシンボリック・リンクを変更します。

```
(
/usr/sbin/xntpd-->
/usr/sbin/ntp4/ntpd4
)
```

1. `xntpd` デーモンを開始するには、次のように入力します。

```
startsrc -s xntpd
```

2. **xntpd** デーモンを停止するには、次のように入力します。

```
stopsrc -s xntpd
```

ファイル

項目	説明
/usr/sbin/ ntp4/ntpd4	ntpd4 デーモンが入っています。 /usr/sbin ディレクトリーから NTP バージョン 3 バイナリーへのデフォルト・シンボリック・リンク。
/etc/ ntp.conf	/usr/sbin/xntpd --> /usr/sbin/ntp3/xntpd デフォルトの構成ファイルが入っています。
/etc/ ntp.drift	デフォルトのドリフト・ファイルが入っています。

関連資料:

- 268 ページの『**ntpq4** デーモン』
- 250 ページの『**ntpdate4** コマンド』
- 263 ページの『**ntpq** コマンド』
- 『**ntpdate** コマンド』
- 272 ページの『**ntptrace** 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。

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、ローカルの日付と時刻を設定します。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一気に修正します。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、サーバーの数が多くな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** 権限を持っていなければなりません。

フラグ

項目	説明
-a <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	settimeofday サブルーチンを呼び出して、クロックの時間を一気に修正します。
-c	adjtime サブルーチンを呼び出して、クロックの時間を一気に修正します。
-d	デバッグ・モードを指定します。 ntpdate 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。
-e <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。
-k <i>KeyFile</i>	デフォルトの /etc/ntp.keys 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、 xntpd コマンドを参照してください。
-o <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。デフォルトは 3 です。
-p <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。
-s	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。 cron コマンドを使用して ntpdate コマンドを実行するときに役に立ちます。
-t <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。
-u	パケットの送信元の非権限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

項目	説明
<i>Server ...</i>	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。

項目	説明
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ません。

イベントの監査: N/A

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

アドレス 9.3.149.107 の NTP サーバーをポーリングしてローカルの日付と時刻を設定するには、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

項目	説明
<code>/usr/sbin/ntpdate</code>	ntpdate コマンドが入っています。
<code>/etc/ntp.keys</code>	デフォルトのキー・ファイルが入っています。

関連資料:

263 ページの『**ntpq** コマンド』

272 ページの『**ntptrace** コマンド』

関連情報:

xntpd コマンド

xntpd コマンド

ntpdate4 コマンド

目的

Network Time Protocol (NTP) を使用して日時を設定します。

構文

```
ntpdate4 [ -4 ] [ -6 ] [ -a key ] [ -B ] [ -b ] [ -d ] [ -e authdelay ] [ -k keyfile ] [ -o version ] [ -p samples ] [ -q ] [ -s ] [ -t timeout ] [ -u ] [ -v ] server [...]
```

説明

ntpdate コマンドは、正確な時刻を決定するためにサーバー引数として指定された Network Time Protocol (NTP) サーバーをポーリングして、ローカルの日付と時刻を設定します。**ntpdate** コマンドは、ローカル・ホスト上で **root** として実行する必要があります。サンプルは指定されている各サーバーから取得され、最良のものを選択するために、クロック・フィルターおよび選択アルゴリズムのサブセットが適用されます。**ntpdate** コマンドの正確性および信頼性は、サーバーの数、このコマンドの実行ごとのポーリングの数、および実行間隔に応じて決まることに注意してください。

ntpdate コマンドは、ホスト・クロックを設定するために必要に応じて手動で実行できます。また、このコマンドは、ブート時にクロックを設定するためにホスト始動スクリプトから実行できます。これは、NTP デーモン **ntpd** を開始する前に初期段階でクロックを設定する場合に役立ちます。**ntpdate** コマンドを **cron** スクリプトから実行することもできます。ただし、不自然な **cron** スクリプトを使用した **ntpdate** コマンドは NTP デーモンの代替にはならないことに留意することが重要です。NTP デーモンは、複雑なアルゴリズムを使用して、リソースの使用を最小限に抑えながら正確性および信頼性を最大化します。最後に、**ntpdate** コマンドは、**ntpd** が行うようにはホスト・クロック周波数を調整しないため、**ntpdate** コマンドを使用した場合の正確性は限定されます。

時間調整は、2 つの方法のうちの 1 つを用いて **ntpdate** コマンドによって行われます。クロックが 0.5 秒を超えたエラー状態であると **ntpdate** コマンドが判断した場合は、**settimeofday()** システム・ルーチン呼び出して時刻を単に一気に修正します。エラー状態が 0.5 秒未満の場合は、**adjtime ()** システム・ルーチン呼び出して時刻を調整します。エラーが小さい場合は後者の手法の方が乱れが少なく、より正確であり、**ntpdate** コマンドが 1 時間または 2 時間ごとに **cron** スクリプトによって実行される場合は有効に働きます。

NTP サーバー・デーモン (**ntpd**) が同じホスト上で実行されている場合は、**ntpdate** コマンドは日付の設定を拒否します。デーモンを実行する代わりに、**cron** スクリプトから定時ベースの **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: ホスト名が必要とされる場合、ホスト名の前に付けられた修飾子 **-4** は、IP バージョン 4 のネーム・スペースに対して DNS 解決を強制します。一方、修飾子 **-6** は、IP バージョン 6 のネーム・スペースに対して DNS 解決を強制します。

フラグ

項目	説明
- 4	IP バージョン 4 のネーム・スペースに対してコマンド・ライン上の以下のホスト名の DNS 解決を強制します。
- 6	IP バージョン 6 のネーム・スペースに対してコマンド・ライン上の以下のホスト名の DNS 解決を強制します。
- a <i>key</i>	認証機能を使用可能にし、引数 keyntpdate として認証に使用されるキー識別子を指定します。キーおよびキー識別子は、クライアントとサーバーの両方の鍵ファイルで一致しなければなりません。デフォルトでは、認証機能は使用不可にされます。
- B	adjtime () システム・コールを使用して時刻を強制的に調整します (測定されたオフセットが + または - 128 ミリ秒を超えている場合でも)。デフォルトでは、オフセットが + または - 128 ミリ秒を超えている場合に、 settimeofday () を使用して時刻を調整します。この場合、オフセットが + または - 128 ミリ秒を大幅に超えていれば、クロックを正しい値に調整するのに長い時間がかかる可能性があります。この時間の間にホストを使用してクライアントを同期化してはなりません。
- b	adjtime () システム・コールを使用して調整 (デフォルト) するのではなく、 settimeofday () システム・コールを使用して時刻を強制的に調整します。ブート時にスタートアップ・ファイルから呼び出された場合にこのオプションを使用する必要があります。
- d	デバッグ・モードを使用可能にします。このモードでは、 ntpdate コマンドはすべてのステップを実行しますが、ローカル・クロックは調整しません。一般デバッグに有効な情報も出力されます。

項目	説明
- e <i>authdelay</i>	認証機能を実行するための処理遅延を、秒単位および 1 秒の何分の一の単位で値 <i>authdelay</i> として指定します (詳しくは、 <i>ntpd</i> を参照してください)。この数値は通常、ほとんどの目的では無視できるほど十分に小さい値です。ただし、値を指定すると、非常に低速の CPU での計時が改善されることがあります。
- k <i>keyfile</i>	認証鍵ファイルのパスを文字列 <i>keyfile</i> として指定します。デフォルトは <i>/etc/ntp.keys</i> です。
- o <i>version</i>	出力パケットの NTP バージョンを整数バージョンとして指定します。これには 1 または 2 を指定できます。デフォルトは 3 です。これにより、 <i>ntpd</i> コマンドを古い NTP バージョンと併用することができます。
- p <i>samples</i>	整数サンプルとして各サーバーから獲得されるサンプルの数を指定します。1 から 8 までの値を使用します。デフォルト値は 4 です。
- q	照会を指定します。クロックは設定しません。
- s	標準出力からのロギング出力を <i>syslog</i> システム・ファシリティに転用します。これは、主として <i>cron</i> スクリプトの便宜を図って設計されています。
- t <i>timeout</i>	サーバーの応答を待つ最大時間を、秒単位および 1 秒の何分の一の単位で値 <i>timeout</i> として指定します。この値は、0.2 秒の倍数に丸められます。デフォルトは 1 秒です。この値は LAN にまたがるポーリングに適しています。
- u	<i>ntpd</i> が非特権ポートまたは出力パケットを使用するように指示します。特権ポートへの着信トラフィックをブロックする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このオプションを使用できます。-d オプションは常に非特権ポートを使用することに注意してください。
- v	詳細出力。このオプションにより、 <i>ntpd</i> バージョン識別文字列がログに記録されます。

パラメーター

項目	説明
<i>Server...</i>	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。

項目	説明
0	正常終了。
>0	エラーが発生しました。

セキュリティ

項目	説明
アクセス制御	このコマンドを実行するには、root 権限を持っている必要があります。
イベントの監査	N/A

例

1. アドレス 9.41.254.24 の NTP サーバーをポーリングしてローカルの日付と時刻を設定するには、下記のように入力します。

```
ntpddate 9.41.254.24
```

下記のような出力が表示されます。

```
address: ::
address: 0.0.0.0
25 Feb 12:19:41 ntpdate[434262]: adjust time server 9.41.254.24 offset -0.005270 sec
```

ファイル

項目	説明
/usr/sbin/ntp4/ ntpdate4	NTP バージョン 4 の ntpdate コマンドが入っています。 /usr/sbin ディレクトリーから NTP バージョン 4 バイナリーへのデフォルト・シンボリック・リンク。 /usr/sbin/ntpdate --> /usr/sbin/ntp3/ntpdate
/etc/ntp.keys	ntpdate コマンドで使用される暗号キー。

関連資料:

- 274 ページの『ntptrace4 コマンド』
- 268 ページの『ntpq4 デーモン』
- 259 ページの『ntp-keygen4 コマンド』
- 263 ページの『ntpq コマンド』

関連情報:

sntp4 コマンド

ntpd4 コマンド

目的

Network Time Protocol (NTP) デーモン **ntpd** の照会または制御プログラムを開始します。

構文

```
ntpd [ -4 ] [-6] [-d] [-i] [-l] [-n] [-p] [-s] [ -c command ] [ host ] [ ... ]
```

説明

ntpd コマンドは、現在の状態について **ntpd** デーモンを照会し、状態の変更を要求するために使用されます。プログラムは対話モードで実行されるか、コマンド・ライン引数を使用して制御される可能性があります。**ntpd** インターフェースから、幅広い状態および統計情報を入手できます。さらに、**ntpd** の構成ファイルを使用して始動時に指定できるすべての構成オプションは、**ntpd** を使用して実行時に指定される場合もあります。

ntpd の実行時に 1 つ以上の要求オプションがコマンド・ラインに含まれている場合、それぞれの要求は、コマンド・ライン引数として指定されている各ホスト上で稼働している NTP サーバー、またはデフォルトでローカル・ホスト上で稼働している NTP サーバーに送信されます。要求オプションが提供されていない場合、**ntpd** は標準入力からコマンドの読み取りを試行し、コマンド・ラインに指定されている最初のホスト上で稼働している NTP サーバー上でこれらを実行します。この場合も、他のホストが指定されていない場合は、ローカル・ホストがデフォルトとなります。標準入力端末が端末デバイスの場合、**ntpd** はコマンドの入力を求めるプロンプトを出します。

ntpd は、NTP モード 7 パケットを使用して NTP サーバーと通信するため、これを許可するネットワーク上の互換性のあるすべてのサーバーを照会するために使用することができます。NTP は UDP プロトコルのため、特にネットワーク接続形態において距離が非常に離れている場合などには、この通信の信頼性が若干低くなることに注意してください。**ntpd** は要求の再送を試行せず、リモート・ホストが適切なタイムアウト時間内に応答しない場合、要求はタイムアウトになります。

ntpd の操作は特定の **ntpd** デーモンのインプリメンテーションに特有で、このバージョンのデーモン、さらに場合によってはいくつかの以前のバージョンのデーモンのみで機能することが予想できます。ローカ

ル・サーバーの状態に影響を与えるリモート **ntpd** プログラムからの要求は認証が必要になります。このためには、リモート・プログラムとローカル・サーバーの両方が共通キーとキー ID を共有している必要があります。

ホスト名が予期されているコンテキストでは、ホスト名の前に **-4** 修飾子を指定すると、IP バージョン 4 ネーム・スペースへの DNS 解決が強制され、**-6** 修飾子を指定すると、IP バージョン 6 ネーム・スペースへの DNS 解決が強制されることに注意してください。

-i または **-n** 以外のコマンド・ライン・オプションを指定すると、指定された照会 (複数可) が、示されたホスト (複数可) に即座に送信されます。その他の場合、**ntpd** は標準入力から対話形式のコマンドの読み取りを試行します。

フラグ

項目	説明
-4	コマンド・ラインの後続ホスト名の DNS 解決を、IP バージョン 4 ネーム・スペースにするよう強制します。
-6	コマンド・ラインの後続ホスト名の DNS 解決を、IP バージョン 6 ネーム・スペースにするよう強制します。
-c command	以下の引数は対話形式コマンドとして解釈され、指定されたホストで実行されるコマンドのリストに追加されます。複数の -c オプションを実行できます。
-d	デバッグ・モードを使用可能にします。このオプションは、複数回使用できます。
-i	ntpd を対話モードで実行することを強制します。プロンプトは、標準入力から読み取られた標準出力およびコマンドに書き込まれます。
-l	サーバーによって認識されるピアのリストを取得します。このスイッチは、 -c listpeers と同等です。
-n	すべてのホスト・アドレスを正規ホスト名に変換するのではなく、ドット 10 進表記数値フォーマットで出力します。
-p	サーバーが認識するピアのリストおよびそれらのピアの状態の要約を表示します。これは、 -c peers と同等です。
-s	サーバーが認識するピアのリストおよびそれらのピアの状態の要約を表示します。表示フォーマットは、 -p スイッチと異なります。これは、 -c dmpeers と同等です。

パラメーター

項目	説明
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。

項目	説明
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

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の

『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

1. Network Time Protocol デモンの照会/制御プログラムを開始するには、次のように入力します。

```
ntpd
```

2. サーバーが認識するピアのリストならびにそれらの状態の要約を表示するには、次のように入力します。

```
ntpd -p
```

下記のような出力が表示されます。

```
remote          local          st  poll reach  delay  offset  disp
=====
ausgsa.austin.ibm.com 9.124.101.190  2  64   1   0.29128 -0.013381 2.81735
```

ntpd 内部コマンド

対話式コマンド

対話形式のコマンドは、1 個のキーワードとその後に続く 0 個から 4 個の引数で構成されます。完全なキーワードのうちコマンドを一意的に識別するのに十分な数の文字のみを入力する必要があります。コマンドの出力は標準出力に通常送信されますが、コマンド・ラインに `<` を付加し、その後にファイル名を指定することにより、オプションで個別のコマンドの出力をファイルに送信することもできます。

多くの対話形式コマンドは、完全に `ntpd` プログラム自体の内部で実行され、結果として NTP モード 7 要求がサーバーに送信されることはありません。以下のリストで、対話式コマンドについて説明します。

項目	説明
? [command_keyword] または help [command_keyword]	疑問符 (?) は単独で、この <code>ntpd</code> のインカーネーションに認識されているすべてのコマンド・キーワードのリストを表示します。疑問符 (?) の後にコマンド・キーワードを指定すると、コマンドの機能と使用法が表示されます。
delay milliseconds	認証を必要とする要求に組み込まれたタイム・スタンプに追加される時間間隔を指定します。これを使用すると、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マシン間で (信頼性の低い) サーバーを再構成することが可能になります。将来の照会の送信先ホストを設定します。 <code>hostname</code> は、ホスト名でも数値アドレスでもかまいません。
host hostname	<code>yes</code> が指定されていると、情報表示画面にホスト名が表示されます。 <code>no</code> が指定されていると、代わりに数値アドレスが表示されます。コマンド・ライン <code>-n</code> スイッチを使用して変更されていない限り、デフォルトは <code>yes</code> です。
hostnames [yes no]	<code>ntpd</code> からホストへの構成要求の認証に使用されるキー番号を指定することができます。これは、ホスト/サーバーがこの目的に使用するように構成されているキー番号に一致する必要があります (サーバー・オプション: <code>trustedkey</code> および <code>requestkey</code>)。 <code>ntpd</code> コマンド用にホスト上で認証が使用可能になっていない場合、コマンド <code>keyid 0</code> を指定する必要があります。指定しないと、次に続く <code>addpeer/addserver/broadcast</code> コマンドの <code>keyid</code> が使用されます。
keyid keyid	<code>ntpd</code> を終了します。
quit	構成要求の認証に使用されるパスワード (エコー出力されません) の入力を求めるプロンプトが出されます。そのような要求が成功するためには、このパスワードが、NTP サーバーによってこの目的のために使用されるように構成されているキーに一致する必要があります。
passwd	

項目	説明
timeout milliseconds	<p>サーバー照会への応答のタイムアウト時間を指定します。デフォルトは、約 8000 ミリ秒です。</p> <p>注: ntpdc は、タイムアウトの後に各照会を 1 回再試行するので、タイムアウトの合計待ち時間は、設定されたタイムアウトの 2 倍になります。</p>

制御メッセージ・コマンド

QUERY コマンドを実行すると、情報の要求を含む NTP モード 7 パケットがサーバーに送信されます。これらは読み取り専用コマンドで、サーバー構成状態が変更されることはありません。

項目	説明
listpeers	サーバーが状態を維持しているピアの短いリストを取得して表示します。このリストには、すべての構成済みピアのアソシエーション、および今後同期化されるとサーバーが解釈する階層をもつピアが含まれています。

peers	<p>サーバーが状態を維持しているピアのリストを、その状態の要約とともに取得します。要約情報には、リモート・ピアのアドレス、ローカル・インターフェース・アドレス (まだローカル・アドレスが決定されていない場合、0.0.0.0)、リモート・ピアの階層 (階層 16 は、そのリモート・ピアが同期化されていないことを示しています)、ポーリング間隔 (秒数)、到達可能レジスター (8 進数)、およびピアの現在予測される遅延、オフセット、分散 (すべて秒数) が含まれています。</p>
--------------	--

左マージンの文字は、このピア・エントリーが作動しているモードを示しています。+ は対称アクティブを示し、- は対称パッシブを示し、= はリモート・サーバーがクライアント・モードでポーリングされていることを意味し、^ はサーバーがこのアドレスにブロードキャストしていることを示し、~ はリモート・ピアがブロードキャストを送信していることを示し、* はサーバーが現在同期化しているピアにマークを付けます。

dmpeers	<p>ホスト・フィールドの内容は、4 つの形式のうち 1 つになる可能性があります。これはホスト名、IP アドレス、パラメーターを指定した参照クロックのインプリメンテーション名、または REFCLK (implementation number, parameter) のいずれかの形式です。hostnames no では、IP アドレスのみが表示されます。</p>
----------------	---

若干異なるピア要約リスト。左端の欄の文字を除き、peers コマンドの出力と全く同じです。文字は、クロック選択アルゴリズムの最終段階に含まれていたピアの横のみ表示されます。「。」は、このピアが疑似ティック検出で切り捨てられたことを示しており、「+」は、このピアが疑似ティック検出を通過したことを示しています。「*」は、サーバーが現在同期化しているピアを示しています。

showpeer peer_address [...]	1 つ以上のピアの現在のピア変数の詳細表示を示します。これらの変数のほとんどは、NTP バージョン 2 仕様で説明されています。
------------------------------------	--

pstats peer_address [...]	指定したピアに関連したピアごとの統計カウンターを表示します。
----------------------------------	--------------------------------

clockinfo clock_peer_address [...]	ピア・クロックに関する情報を取得して表示します。取得した値は、誤差の設定に関する情報およびその他のクロック・パフォーマンス情報を提供します。
---	--

kerninfo	カーネル・フェーズ・ロック・ループ操作パラメーターを取得します。この情報は、カーネルが正確な計時機能を持つように特別に変更された場合のみ使用可能です。
-----------------	---

loopinfo [oneline multiline]	<p>選択されたループ・フィルター変数の値を表示します。ループ・フィルターは、ローカル・システム・クロックを調整する NTP の一部です。offset は、パケット処理コードによってループ・フィルターに最後に指定されるオフセットです。frequency は、ローカル・クロックの周波数誤差を百万分率 (ppm) で表したものです。time_const は、フェーズ・ロック・ループの強固さ、およびそれが発振器ドリフトに適応できるスピードを制御します。watchdog timer の値は、ループ・フィルターにサンプル・オフセットが最後に指定されてから経過した秒数です。oneline オプションと multiline オプションは、この情報を表示するフォーマット (デフォルトは multiline) を指定します。</p>
---	--

項目	説明
sysinfo	ローカル・サーバーに関連するさまざまなシステム状態変数を表示します。最後の 4 行を除くすべてが NTP バージョン 3 仕様、RFC-1305 で説明されています。
	system flags はさまざまなシステム・フラグを示し、そのいくつかは enable 構成コマンドによって設定し、 disable 構成コマンドによってクリアすることができます。これらは、 auth 、 bclient 、 monitor 、 pll 、 pps 、および stats フラグです。これらのフラグの意味については、 ntpd の資料を参照してください。この他に、2 つの読み取り専用フラグ kernel_pll と kernel_pps があります。これらのフラグは、正確な時間のためのカーネル変更が行われているときの同期化状態を示します。 kernel_pll は、ローカル・クロックがカーネルによって統制されていることを示し、 kernel_pps はカーネルの統制が PPS 信号によって提供されていることを示します。
	stability は、システム頻度を訂正した後に残った残余頻度エラーで、保守とデバッグに使用されます。ほとんどのアーキテクチャーでは、最初にこの値は 500 ppm の高い値から、.01 から 0.1 ppm の範囲の公称値に減少します。デーモンの開始後もしばらくこの値が高いままであれば、ローカル・クロックに異常があるか、カーネル変数の値 tick が正しくない可能性があります。
	broadcastdelay は、 broadcastdelay 構成コマンドによって設定されたデフォルトのブロードキャストの遅延を示します。
sysstats	authdelay は、 authdelay 構成コマンドによって設定されたデフォルトの認証の遅延を示します。
memstats	プロトコル・モジュールで保守される統計カウンターを表示します。
iostats	メモリー割り当てコードに関連する統計カウンターを表示します。
timerstats	入出力モジュールで保守される統計カウンターを表示します。
reslist	タイマー/イベント・キュー・サポート・コードで保守される統計カウンターを表示します。
	サーバーの制限事項リストを取得して表示します。このリストはソート順に表示され、制約事項の適用方法に理解に役立つ可能性があります。
ifstats	ntpd がネットワーク通信に使用するインターフェースのインターフェース統計をリストします。
ifreload	現在のシステム・インターフェースのスキャンを強制します。変更される可能性のあるインターフェースのインターフェース統計を出力します。変更されていないインターフェースに「.」、追加されたインターフェースに「+」、削除されたインターフェースに「-」のマークを付けます。
monlist [version]	モニター機能によって収集され、保守される通信量カウントを取得して表示します。通常、バージョン番号を指定する必要はありません。
clkbug clock_peer_address [...]	参照クロック・ドライバーのデバッグ情報を取得します。この情報はいくつかのクロック・ドライバーによってのみ提供され、ドライバー・ソースのコピーがないとデコードできません。

ランタイム構成要求

サーバーの状態変更を引き起こすすべての要求は、構成された NTP キーを使用して、サーバーによって認証されます (サーバーは、キーを構成しないことにより、この機能を使用不可にすることもできます)。また、キー番号と対応するキーが、**ntpd** に認識されている必要があります。これは、**keyid** コマンドと **passwd** コマンドを使用して実行できます。後者のコマンドを実行すると、暗号キーとして使用するパスワードの入力を求めるプロンプトが端末で表示されます。また、結果的にサーバーへの認証要求を出すコマンドを初めて使用すると、自動的にキー番号とパスワードの両方の入力を求めるプロンプトが表示されます。認証は、要求側がそのような変更を行う権限を持っているかを確認するだけでなく、伝送エラーに対して、より高い保護も提供します。

認証要求は必ずパケット・データにタイム・スタンプを組み込み、これは認証コードの計算に組み込まれます。このタイム・スタンプはサーバーにより、受信タイム・スタンプと比較されます。この時間差が少量を超えていた場合、要求は拒否されます。これは、2 つの理由により行われます。1 つ目は、LAN 上で通信内容を傍受できる可能性のある者による、サーバーでの単純なりプレイ・アタックをより困難にします。2 つ目は、接続形態上リモートにあるホストからサーバーに構成変更を要求することをより困難にします。再構成機能は、ローカル・ホスト上のサーバーではうまく動作し、同じ LAN 上の時間的に同期されたホスト間では適切に動作する可能性があります。さらに離れたホストでは十分に動作しません。したがっ

て、合理的なパスワードを選択し、キーの配布および保護に気を付け、送信元アドレスに適切な制約を適用すれば、実行時の再構成機能によってセキュリティの水準が十分に確保されるはずで

下記のコマンドは、認証要求を実行します。

項目	説明
<code>addpeer peer_address [keyid] [version] [minpoll# prefer iburst burst minpoll N maxpoll N [...]]</code>	対称アクティブ・モードで操作中の構成されたピアのアソシエーションを指定のアドレスに追加します。このコマンドを実行すると、必要に応じて、同じピアをもつ既存のアソシエーションが削除されたり、単に新規構成に適合するように変換されたりする可能性があることに注意してください。 keyid がゼロ以外の場合、リモート・サーバーへのすべての出力パケットに、このキーによって暗号化された認証フィールドが付加されます。値が 0 の場合 (または指定されていない場合)、認証は行われません。 ntpdc のキー番号が (例えば、 keyid コマンドによって) まだ設定されていない場合は、この値に設定されます。 version# は 1 から 4 の値を取ることができ、デフォルト値は 3 です。残りのオプションは、 minpoll の数値か、リテラル prefer 、 iburst 、 burst 、 minpoll N 、 keyid N 、 version N 、または maxpoll N (ここで、N は数値) で、ピア構成ファイル・コマンド ntpd に指定されたアクションを実行します。それぞれのフラグが指定される (または指定されない) ことにより、前の設定が置き換えられます。 prefer キーワードは優先ピアを示します (したがって、可能な場合、主にクロックの同期に使用されます)。また、優先ピアは PPS 信号の妥当性も判別します。優先ピアが同期化に適している場合、PPS 信号にもそれが当てはまります。動作モードがクライアントであることを除き、 addpeer コマンドと全く同じです。
<code>addserver peer_address [keyid] [version] [minpoll# prefer iburst burst minpoll N maxpoll N [...]]</code>	
<code>addserver peer_address [prefer iburst burst minpoll N maxpoll N keyid N version N [...]]</code>	
<code>broadcast peer_address [keyid] [version] [prefer]</code>	動作モードがブロードキャストであることを除き、 addpeer コマンドと全く同じです。この場合、有効なゼロ以外のキー ID とキーが必要です。 peer_address パラメーターは、ローカル・ネットワークのブロードキャスト・アドレス、または NTP に割り当てられたマルチキャスト・グループ・アドレスの場合があります。マルチキャスト・アドレスの場合は、マルチキャスト対応のカーネルが必要です。
<code>unconfig peer_address [...]</code>	このコマンドを実行すると、指定されたピアから構成されたビットが除去されます。多くの場合、これによりピアのアソシエーションが削除されます。ただし、リモート・ピアをこの方法で継続させたいときなど、場合によっては、アソシエーションを構成解除されたモードで持続させてもかまいません。
<code>fudge peer_address [time1] [time2] [stratum] [refid]</code> <code>enable [auth bclient calibrate kernel monitor ntp pps stats]</code>	このコマンドは、参照クロックの特定のデータを設定する方法を提供します。詳しくは、ソース・リストを参照してください。これらのコマンドは、 ntpd の enable および disable 構成ファイル・コマンドと同じ方法で作動します。
<code>disable [auth bclient calibrate kernel monitor ntp pps stats]</code>	
<code>restrict address mask flag [flag]</code> <code>unrestrict address mask flag [flag]</code> <code>delrestrict address mask [ntpport]</code> <code>readkeys</code>	このコマンドは、 ntpd の restrict 構成ファイル・コマンドと同じ方法で作動します。制約事項リストからマッチング・エントリーの制約事項を除去します。制限事項リストからマッチング・エントリーを削除します。現在の認証キーのセットを消去し、キー・ファイル (これは、 ntpd 構成ファイルに指定されている必要があります) を再度読み取って新規セットを取得します。これにより、サーバーを再始動せずに暗号キーを変更することができます。
<code>trustedkey keyid [...]</code>	これらのコマンドは、 ntpd の trustedkey および untrustedkey 構成ファイル・コマンドと同じ方法で作動します。
<code>untrustedkey keyid [...]</code> <code>authinfo</code>	認識されたキーと実行された暗号化および暗号化解除のカウントを含む認証モジュールに関する情報を戻します。
<code>traps</code>	サーバーに設定されたトラップを表示します。詳しくは、ソース・リストを参照してください。
<code>addtrap [address [port] [interface]</code>	非同期メッセージングのトラップを設定します。詳しくは、ソース・リストを参照してください。

項目	説明
<code>clrtrap [address [port] [interface]</code>	非同期メッセージングのトラップをクリアします。詳しくは、ソース・リストを参照してください。
<code>reset</code>	サーバーのさまざまなモジュールの統計カウンターをクリアします。詳しくは、ソース・リストを参照してください。

ファイル

項目	説明
<code>/usr/sbin/ntp4/ntpd4</code>	<code>ntpd</code> コマンドが入っています。 <code>/usr/sbin</code> ディレクトリーから NTP バージョン 3 バイナリーへのデフォルト・シンボリック・リンク。 <code>/usr/sbin/ntpd --> /usr/sbin/ntp3/xntpd</code>

関連資料:

- 250 ページの『`ntpdate4` コマンド』
- 268 ページの『`ntpq4` デーモン』
- 248 ページの『`ntpdate` コマンド』
- 272 ページの『`ntptrace` コマンド』

関連情報:

`xntpd` コマンド

ntp-keygen4 コマンド

目的

公開鍵と秘密鍵を生成します。

構文

```
ntp-keygen [ -d ] [ -e ] [ -G ] [ -g ] [ -H ] [ -I ] [ -M ] [ -P ] [ -T ] [-c [RSA-MD2 | RSA-MD5 |
RSA-SHA | RSA-SHA1 | RSA-MDC2 | RSA-RIPEMD160 | DSA-SHA | DSA-SHA1 ] ] [ -i name ] [
-m modulus ] [ -p password ] [ -q password ] [ -S [ RSA | DSA ] ] [ -s name ] [ -v nkeys ] [ -V
params]
```

説明

`ntp-keygen4` コマンドは、NTP バージョン 4 の認証および識別スキームによって使用される暗号データ・ファイルを生成します。このコマンドは、対称鍵暗号方式で使用される MD5 鍵ファイルを生成します。さらに、OpenSSL ソフトウェア・ライブラリーがインストールされている場合は、公開鍵暗号方式で使用される鍵、証明書、および ID ファイルが生成されます。これらのファイルは、インターネット標準セキュリティ・インフラストラクチャーと互換性がある Cookie 暗号化、デジタル署名、およびチャレンジ/応答識別アルゴリズムに使用されます。

デフォルトでは、ファイルは `ntp-keygen` によって暗号化されません。`-p password` オプションは書き込みパスワードを指定し、`-q password` オプションは前に暗号化されたファイルに関する読み取りパスワードを指定します。`ntp-keygen` プログラムが暗号化されたファイルを読み取り、かつパスワードが欠落しているか誤っている場合は、このプログラムはパスワードの入力を求めます。暗号化されたファイルが正常に読み取られ、さらに書き込みパスワードが指定されない場合は、読み取りパスワードがデフォルトにより書き込みパスワードとして使用されます。

ntpd 構成コマンドの **pw** 暗号パスワードは、前に暗号化されたファイルの読み取りパスワードを指定します。パスワードが欠落しているか誤っている場合は、このデーモンはそのスポットで有効期限が切れます。便宜上、ファイルが前に暗号化されている場合は、デフォルトの読み取りパスワードはプログラムを実行しているホストの名前です。前の書き込みパスワードがホスト名として指定されている場合は、これらのファイルは、明示パスワードを持たないそのホストによって読み取ることができます。

すべてのファイルは PEM エンコード化印刷可能 ASCII フォーマットであるため、他のサイトおよび認証局に対してメール内の MIME 添付ファイルとして組み込むことができます。ファイル名は接頭部 **ntpkey_** から始まり、接尾部 **_hostname.filestamp** で終わります。ここで、**hostname** は通常、UNIX **gethostname()** ルーチンによって戻される文字列であり、**filestamp** はファイル生成時の NTP 秒 (10 進数) です。この両方は固有性を保証し、保守手順を単純化します。これは、すべてのファイルを **rm ntpkey*filestamp** コマンドによって迅速に除去できるか、または特定の時刻に生成されるすべてのファイルを **rm *filestamp** コマンドによって除去できるためです。誤った構成のリスクをさらに削減するために、ファイルの最初の 2 行には、コメントとしてファイル名および生成日時が含まれます。

すべてのファイルは、デフォルトでは、キー・ディレクトリー **/usr/local/etc** にインストールされます。このディレクトリーは、通常、NFS マウント済みネットワークの共有ファイルシステム内にあります。キー・ディレクトリーおよび各ファイルの実際のロケーションは、構成コマンドによってオーバーライドできますが、この処理はお勧めできません。通常、ホストごとのファイルはそのホストによって生成され、そのホストによってのみ使用されます。ただし、このページの後に示されているように、例外が存在します。

通常、専用値 (ホスト・キー、記号キー、および識別パラメーターを含む) が入っているファイルは、**root** 読み取り/書き込み専用として許可されます。一方、公用値が入っている他のファイルは、全ユーザー読み取り可能なものとして許可されます。また、専用値が入っているファイルは暗号化することができ、これらのファイルは全ユーザー読み取り可能なものとして許可されます。これにより、共有ファイルシステムでの保守が単純化されます。固有性はホスト名およびファイル名拡張子によって保証されるため、NFS サーバーおよび従属クライアントのファイルはすべて、同じ共有ディレクトリーにインストールすることができます。

ファイルのインストール時にファイル名拡張子を保持し、このページの他の部分で指定されている総称名から生成済みのファイルへソフト・リンクをインストールすることをお勧めします。これにより、このリンクを変更して新規ファイル生成を簡単に活動化することができます。リンクが存在する場合、**ntpd** は、ファイル・スタンプを抽出するために、ファイル名に合わせてこのリンクに従います。リンクが存在しない場合は、**ntpd** は該当のファイル自体からファイル・スタンプを抽出します。これにより、クライアントは、ファイルおよび生成時刻が常に現行のものであることを確認することができます。**ntp-keygen** プログラムは、同時に生成されたすべてのファイルに同じ拡張子を使用するので、それぞれの生成は別個であり、データのモニター時に容易に認識することができます。

プログラムの実行

ntp-keygen プログラムを実行する最も安全な方法は、**root** として直接ログインすることです。お勧めする手順は、キー・ディレクトリー (通常は **/usr/local/etc**) に変更してからプログラムを実行することです。初めて実行する場合、またはすべての **ntpkey** ファイルが除去された場合は、プログラムは、RSA ホスト鍵ファイルおよび適合する RSA-MD5 証明書ファイルを生成します。多くの場合、必要となるのはこれだけです。プログラムは、総称名からそれぞれのファイルへのソフト・リンクも生成します。再度実行された場合、プログラムは同じホスト鍵ファイルを使用しますが、新規の証明書ファイルおよびリンクを生成します。

ホスト・キーは、必要に応じて Cookie の暗号化に使用されるため、RSA タイプでなければなりません。デフォルトでは、ホスト・キーは、シングニチャーの暗号化に使用される記号キーでもあります。必要なとき

に、別の記号キーを指定することができます。この記号キーは RSA または DSA のいずれのタイプでも構いません。デフォルトでは、メッセージ・ダイジェスト・タイプは MD5 ですが、OpenSSL ライブラリーによってサポートされる記号キー・タイプとメッセージ・ダイジェスト・タイプを任意に組み合わせたものを指定できます。これには、MD2、MD5、SHA、SHA1、MDC2、および RIPE160 メッセージ・ダイジェスト・アルゴリズムを使用したものも含まれます。ただし、証明書に指定されているスキームは記号キーと互換性がなければなりません。任意のダイジェスト・アルゴリズムを使用する証明書は、RSA 記号キーと互換性があります。ただし、SHA および SHA1 証明書のみが DSA 記号キーと互換性があります。

秘密/公開鍵ファイルおよび証明書は、他の OpenSSL アプリケーションと互換性があります。また、多くの場合、他のライブラリーとも同様に互換性があります。証明書およびその証明書から派生した認証要求は、現存の業界慣行と互換性がなければなりません。一部のユーザーは、X509v3 拡張フィールドの解釈にある程度の自由度があることを認識しています。ただし、識別パラメーター・ファイルはその他のファイルとしてエンコードされますが、Autokey 以外のものとはおそらく互換性がありません。

root 以外のユーザーとしてプログラムを実行し、UNIX su コマンドを使用して root の役割を担っても、適切に働かないことがあります。これは、デフォルトでは、OpenSSL ライブラリーがユーザー・ホーム・ディレクトリー内のランダム・シード・ファイル .rnd を探すためです。ただし、ルート・ディレクトリー内では都合なことに .rnd ファイルは 1 つしかないはず。そのため、OpenSSL ライブラリーによって使用される \$RANDFILE 環境変数を /.rnd へのパスとして定義するのが便利です。

root としてキーをインストールしても、NFS マウント済み共有ファイルシステムでは機能しない可能性があります。これは、NFS クライアントがたとえ root であっても共有キー・ディレクトリーに書き込みを行うことができない場合があるからです。この場合、NFS クライアントは、keydir コマンドを使用して /etc などの別のディレクトリーにファイルを指定できます。あるクライアントが他のクライアントまたはサーバーのキーおよび証明書を読み取る必要はありません。これは、これらのデータが Autokey プロトコルによって自動的に取得されるためです。

通常、暗号ファイルはそれらを使用するホストによって生成されますが、トラステッド・エージェント (TA) が他のホスト用にこれらのファイルを生成することができます。ただし、このような場合、ファイルは常に暗号化する必要があります。サブジェクト名とトラステッド名のデフォルトは、ファイルを生成するホストのホスト名になりますが、コマンド・ライン・オプションによって変更することができます。所有者名とトラステッド名をそれぞれ、証明書のサブジェクト・フィールドと送出者フィールドとして指定するのが便利です。所有者名はホストおよび記号キー・ファイルにも使用されます。一方、トラステッド名は ID ファイルに使用されます。

フラグ

項目	説明
-c [RSA-MD2 RSA-MD5 RSA-SHA RSA-SHA1 RSA-MDC2 RSA-RIPEMD160 DSA-SHA DSA-SHA1]	証明書メッセージ・ダイジェスト/シグニチャー暗号化スキームを選択します。RSA スキームは RSA 記号キーと併用する必要があり、DSA スキームは DSA 記号キーと併用する必要があります。このオプションを指定しないデフォルトは RSA-MD5 です。
-d	デバッグを使用可能にします。このオプションは、見やすい掲示板で生成される暗号データを表示します。
-e	IFF クライアント・キーを標準出力に書き込みます。これは、メールによる自動キー配布を対象としています。
-G	GQ 識別スキーム用のパラメーターおよびキーを生成し、既存のものがあればそれと差し替えます。
-g	既存の GQ パラメーターを使用して GQ 識別スキームのキーを生成します。GQ パラメーターが存在していない場合は、最初にこのパラメーターを作成します。
-H	新規ホスト・キーを生成し、既存のものがあればそれと差し替えます。
-I	IFF 識別スキーム用のパラメーターを生成し、既存のものがあればそれと差し替えます。

項目	説明
-i name	サブジェクト名を name に対して設定します。これは、証明書内およびホスト・キーと記号キーのファイル名の中のサブジェクト・フィールドとして使用されます。
-M	MD5 キーを生成し、既存のものがあればそれと差し替えます。
-m modulus	プライム・モジュラス・サイズをビット単位 (256 - 2048) で設定します。デフォルト・サイズは 512 です。
-P	プライベート証明書を生成します。デフォルトでは、このプログラムはパブリック証明書を生成します。
-p password	パスワードを含んだ専用データが入っている生成済みファイルおよび DES-CBC アルゴリズムを暗号化します。
-q password	ファイルを読み取るためのパスワードを password に対して設定します。
-S [RSA DSA]	指定されたタイプの新規記号キーを生成し、既存のものがあればそれと差し替えます。デフォルトでは、このプログラムはホスト・キーを記号キーとして使用します。
-s name	送付者名を name に対して設定します。これは、証明書内および ID ファイルのファイル名の中の送付者フィールドとして使用されます。
-T	トラステッド証明書を生成します。デフォルトでは、このプログラムは非トラステッド証明書を生成します。
-V nkeys	Mu-Varadharajan (MV) 識別スキーム用のパラメーターおよびキーを生成します。

終了状況

このコマンドは、以下の終了値を戻します。

項目	説明
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. RSA-SHA 暗号キーを生成するには、次のように入力します。

```
ntp-keygen -c RSA-SHA
```

2. サーバーが認識するピアのリストならびにそれらの状態の要約を表示するには、次のように入力します。

```
ntpd -p
```

下記のような出力が表示されます。

```
Using OpenSSL version 90804f
Generating RSA keys (512 bits)...
RSA                               3 1 2
Generating new host file and link
ntpkey_host_aixfvt12->ntpkey_RSAkey_aixfvt12.3444540821
Using host key as sign key
Generating certificate RSA-SHA
X509v3 Basic Constraints: critical,CA:TRUE
```



```
X509v3 Key Usage: digitalSignature,keyCertSign
Generating new cert file and link
ntpkey_cert_aixfvt12->ntpkey_RSA-SHAcert_aixfvt12.3444540821
```

ファイル

項目	説明
<code>/usr/sbin/ntp4/ntp-keygen4</code>	ntp-keygen コマンドが入っています。 <code>/usr/sbin</code> ディレクトリーから NTP バージョン 4 バイナリーへのデフォルト・シンボリック・リンク。 <code>/usr/sbin/ntp-keygen --> /usr/sbin/ntp4/ntp-keygen4</code>

関連資料:

250 ページの『`ntpdate4` コマンド』

268 ページの『`ntpq4` デーモン』

『`ntpq` コマンド』

関連情報:

`sntp4` コマンド

`xntpd` コマンド

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。

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・ライン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ます、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

項目	説明
-c <i>SubCommand</i>	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 <i>SubCommand</i> 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

項目	説明
<i>Host</i> ...	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。

項目	説明
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

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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・ラインに付加し、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。

項目	説明
? [<i>SubCommand</i>]	コマンド使用情報を表示します。 <i>SubCommand</i> 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。 <i>SubCommand</i> 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。
addvars <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、リストをコマンドで区切り、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。
authenticate yes no	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、 ntpq 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。
clearvars	すべての変数を内部データ・リストから除去します。
cooked	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。
debug more less off	ntpq 照会プログラムのデバッグをオンまたはオフにします。 more および less 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。
delay <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。
host <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。
hostnames yes no	ホスト名を出力するか (yes) または数値アドレスを出力するか (no) を指定します。 -n フラグが使用されない場合は、 yes 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。
keyid <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。
ntpversion 1 2 3	パケットをボ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。 注: 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。
passwd	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。
quit	ntpq 照会プログラムを終了します。
raw	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。
rmvars <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、リストをコマンドで区切り、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。
timeout <i>MilliSeconds</i>	サーバー照会への応答のタイムアウト時間を指定します。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。 注: ntpq 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。ピア変数を通じた 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。

項目	説明
associations	<p>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。</p> <ul style="list-style-type: none"> • 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。 <p>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。</p>
clockvar [<i>AssocID</i>] [<i>Variable</i> [= <i>Value</i>], ...] or cv [<i>AssocID</i>] [<i>Variable</i> [= <i>Value</i>], ...]	<p>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、<i>AssocID</i> 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。</p>
lassociations	<p>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。</p>
lpassociations	<p>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。</p>
lpeers	<p>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。</p>
mreadvar <i>AssocID AssocID</i> [<i>Variable</i> [= <i>Value</i>], ...] or mrvar <i>AssocID AssocID</i> [<i>Variable</i> [= <i>Value</i>], ...]	<p>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。</p>
mreadlist <i>AssocID AssocID</i> or mrl <i>AssocID AssocID</i>	<p>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。</p>
opeers	<p>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。</p>
passociations	<p>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。</p>

項目
peers

説明

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カーが指定されました。
- . 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*

状況の読み取り要求を送信して、指定さ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。

readlist [*AssocID*] または **rl** [*AssocID*]

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、*AssocID* 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。

readvar [*AssocID*] [*Variable* [=*Value*], ...] or **rv** [*AssocID*] [*Variable* [=*Value*], ...]

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、*AssocID* 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を戻します。

writevar [*AssocID*] [*Variable* [=*Value*], ...]

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。

writelist [*AssocID*]

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

項目	説明
<code>/usr/sbin/ntpq</code>	<code>ntpq</code> コマンドが入っています。

関連資料:

- 248 ページの『`ntpdate` コマンド』
- 272 ページの『`ntptrace` コマンド』

関連情報:

- `xntpdc` コマンド
- `xntpd` コマンド

ntpq4 デーモン

目的

標準の Network Time Protocol (NTP) 照会プログラムを始動します。

構文

```
ntpq [-4 -6 -d -i -n -p] [-c command] [host] [...]
```

説明

`ntpq` プログラムは、NTP デーモンおよび `ntpd` 操作をモニターし、パフォーマンスを判別するために使用します。このプログラムは、RFC 1305 によって定義された標準 NTP バージョン 3 モード 6 制御メッセージ・フォーマットを使用します。同じフォーマットは NTP バージョン 4 で使用されます。

このプログラムは、コマンド・ライン引数を使用して対話モードまたは制御モードのいずれかで実行することができます。ロー出力および印刷出力のオプションを使用して、任意の変数の読み取りおよび書き込みを行うための要求をアセンブルすることができます。`ntpq` プログラムは、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。

`ntpq` プログラムの実行時に 1 つ以上の要求オプションがコマンド・ラインに組み込まれている場合は、各要求は、コマンド・ライン引数によって指定されたホスト上またはデフォルトのローカル・ホスト上で実行する NTP サーバーに送信されます。要求オプションを何も指定しないと、`ntpq` ユーティリティーは、コマンド・ラインで最初に指定されたホストで実行する 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ます。また、他のいずれのホストも指定しない場合は、ローカル・ホストがデフォルトにされます。標準入力端末デバイスの場合は、`ntpq` ユーティリティーは、コマンドの入力を求めません。

`ntpq` ユーティリティーは、NTP モード 6 のパケットを使用して NTP サーバーと通信し、さらにこのユーティリティーを使用して、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のサーバーに照会することができます。

ホスト名が必要とされるインスタンスで、ホスト名の前に修飾子 `-4` を追加する場合は、このユーティリティーは、IP バージョン 4 のネーム・スペースに対して DNS 解決を強制します。同様に、修飾子 `-6` を使用すると、IP バージョン 6 のネーム・スペースに対して DNS 解決が強制されます。

-i または **-n** 以外のコマンド・ライン・オプションを指定すると、指定された照会が、指示されたホストにただちに送信されます。そうでない場合は、**ntpq** ユーティリティーは、標準入力から対話形式のコマンドを読み取ろうとします。

フラグ

項目	説明
-4	IP バージョン 4 のネーム・スペースに対してコマンド・ライン上のホスト名の DNS 解決を強制します。
-6	IP バージョン 6 のネーム・スペースに対してコマンド・ライン上のホスト名の DNS 解決を強制します。
-c	次の引数は対話形式のコマンドとして解釈され、指定されたホストで実行されるコマンドのリストに追加されます。複数の -c オプションが追加されることもあります。
-d	デバッグ・モードを使用可能にします。
-i	ntpq ユーティリティーを対話モードで動作するように強制します。この結果は標準出力に書き込まれ、コマンドは標準入力から読み取られます。
-n	標準ホスト名に変換するのではなく、すべてのホスト・アドレスをドット 10 進表記数値フォーマットで出力します。
-p	サーバーが認識するピアのリストならびにそれらの状態の要約を表示します。これはピア対話式コマンドと同等です。

パラメーター

項目	説明
<i>Host...</i>	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、**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

- 対話モードで Network Time Protocol の照会プログラムを開始するには、次のように入力します。

```
ntpq -i
```

- サーバーが認識するピアのリストならびにそれらの状態の要約を表示するには、次のように入力します。

```
ntpq -p
```

以下のような出力が表示されます。

```
remote          refid          st t  when poll reach  delay    offset    jitter
=====
ausgsa.austin.ibm.com 9.41.253.167  2 u  19    64 377    285.962  -8.792   2.989
```

ntpq 内部コマンド

対話形式のコマンド

対話形式のコマンドは、1 個のキーワードとその後に続く最大 4 個の引数で構成されます。コマンドを一意的に識別するために、キーワードの文字を必要数だけ入力する必要があります。コマンドの出力は通常、標準出力に送信されます。個々のコマンドの出力は、より大記号 (>) の後にファイル名を付けてコマンド・ラインに付加して送信することもできます。対話形式のコマンドには、完全に **ntpq** 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あります。

項目	説明
? [command_keyword] または help [command_keyword]	疑問符 (?) は単独で、 ntpq のこの具体化によって認識されるすべてのコマンド・キーワードのリストを表示します。疑問符 (?) の後にコマンド・キーワードを指定すると、コマンドの機能と使用法が表示されます。
addvars <i>variable_name</i> [= <i>value</i>] [...] または rmvars <i>variable_name</i> [...] または clearvars	NTP モード 6 のメッセージで送られるデータは、形式 <i>variable_name</i> = <i>value</i> の項目リストで構成されます。ここでは、変数を読み取るサーバーへの要求に応じて、等号 (=) 値は無視されるので、省略することができます。 ntpq プログラムは、内部リストを保持しますが、そのリスト内では下記の readlist および writelist 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センブルし送信することができます。 addvars コマンドには、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できません。複数の変数を追加する場合は、リストをコンマで区切る必要があります、空白文字を含んではなりません。 rmvars コマンドを使用してリストから個々の変数を除去できます。一方、 clearlist コマンドはリストからすべての変数を除去します。
cooked	QUERY コマンドからの出力が生成されます。その結果、 ntpq コマンドによって認識される変数には、人間が使用するために再フォーマットされた値が含まれます。 ntpq プログラムは、変数値をデコードできない場合に、末尾に疑問符 (?) を使用して変数にマーク付けします。
debug <i>more</i> <i>less</i> <i>no</i> delay <i>milliseconds</i>	ntpq デバッグのレベルを調整します。デフォルトは debug no です。認証を必要とする要求に組み込まれたタイム・スタンプに追加される時間間隔を指定します。これにより、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マシン間でサーバーを再構成することができます。
host <i>hostname</i>	将来の照会の送信先ホストを設定します。 <i>hostname</i> は、ホスト名でも数値アドレスでもかまいません。
hostnames [<i>yes</i> <i>no</i>]	yes を指定すると、ホスト名が情報表示画面に表示されます。 no を指定すると、数値アドレスが表示されます。コマンド・ライン -n スイッチを使用して変更されていない限り、デフォルトは yes です。
keyid <i>keyid</i>	構成要求の認証に使用するキー番号を指定します。これは、この目的のためにサーバーが使用するように構成されたキー番号に対応する必要があります。
ntpversion <i>1</i> <i>2</i> <i>3</i> <i>4</i>	ntpq がパケットで要求する NTP バージョン番号を設定します。デフォルトは 2 です。モード 6 の制御メッセージは、NTP バージョン 1 にはありませんでした。
passwd	エコー出力されず、かつ構成要求の認証に使用されるパスワードの入力を求めます。このパスワードは、この目的のために NTP サーバー用に構成されたキーに対応する必要があります。
quit	ntpq を終了します。
raw	リモート・サーバーから受信した QUERY コマンドの出力を表示します。このデータに関して行われた唯一のフォーマット設定により、非 ASCII データを印刷可能形式に変換します。
timeout <i>milliseconds</i>	サーバー照会への応答のタイムアウト時間を指定します。デフォルトは 5000 ミリ秒です。 ntpq 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・コマンド

NTP サーバーが認識する各アソシエーションには、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、対応するピアの値を識別しなければなりません。こ

これらのピアの値はそのアソシエーション ID です。0 のアソシエーション ID は、その変数がシステム変数であり、その名前が個別のネーム・スペースから取り出されたものであることを示します。

制御メッセージ・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** 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** コマンドがあります。

項目	説明
associations	<p>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。リストは列に表示されます。</p> <p>最初の列は、1 を先頭にしてアソシエーションの索引番号付けを示します。2 番目の列は、サーバーによって戻された実際のアソシエーション ID を示し、3 番目の列はピアの状況ワードを示します。これらの列の後には、状況ワードからデコードしたデータが入っているいくつかの列が続きます。状態フィールドのデコードについては、peers コマンドを参照してください。</p> <p>注:</p> <ol style="list-style-type: none"> 1. associations コマンドによって戻されたデータは、ntpq 内に内部的にキャッシュされます。 2. &index 形式の索引は、アソシエーション ID を使用するサーバーを扱うときに使用されます。この場合、後続のコマンドには、引数としてアソシエーション ID が必要となります。
clockvar [assocID] [variable_name [= value [...]] [...]	<p>サーバーのクロック変数のリストを送信するようにサーバーに要求します。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。アソシエーション ID が省略されるかまたはゼロの場合は、システム・クロック変数を要求しており、クロックを持つすべてのサーバーから肯定応答を得ます。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。</p>
cv [assocID] [variable_name [= value [...]] [...]	<p>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取得して表示します。このコマンドは、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 コマンドと異なります。このようなアソシエーションは通常、associations コマンドの使用時に表示から省略されますが、lassociations の出力に組み込まれます。</p>
lassociations	<p>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。このコマンドは、fuzzball を処理する場合にのみ passociations と異なります。</p>
lpassociations	<p>サーバーが保持しているすべてのアソシエーションの状態の要約が表示されることを除き、R ピアに類似しています。これは、fuzzball 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。</p>
lpeers	<p>照会が (ゼロ以外の) アソシエーション ID の範囲に対して行われることを除き、readlist コマンドに類似しています。最新の associations コマンドでキャッシュしたアソシエーション・リストが、この範囲を決定します。</p>
mreadlist assocID assocID	<p>照会が (ゼロ以外の) アソシエーション ID の範囲に対して行われることを除き、readvar コマンドに類似しています。最新の associations コマンドでキャッシュしたアソシエーション・リストが、この範囲を決定します。</p>
mreadvar assocID assocID [variable_name [= value[...]]	<p>ローカル・インターフェース・アドレスによって置き換えられる参照 ID を指定した古い形式の peers コマンド。</p>
mrvar assocID assocID [variable_name [= value[...]]	<p>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 コマンドと同じように動作します。</p>
opeers	
passociations	

項目	説明
peers	サーバーの現行ピア・リストおよび各ピアの状態の要約を取得します。要約情報には、リモート・ピアのアドレス、参照 ID (不明の場合は 0.0.0.0)、リモート・ピアの階層、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)、最終パケットの受信時刻、ポーリング間隔 (秒単位)、到達可能性レジスター (8 進数)、ならびに、ピアの現在予測される遅延、オフセット、および分散 (すべてミリ秒単位) が含まれます。
pstatus assocID	状況の読み取り要求を、指定されたアソシエーションのサーバーに送信します。戻されたピア変数の名前および値が表示されます。ヘッダーからの状況ワードが、16 進数かつピジン英語で変数の前に表示されることに注意してください。
readlist [assocID]	内部変数リスト内の変数の値がサーバーによって戻されるように要求します。アソシエーション ID が省略されるかまたは 0 の場合、変数はシステム変数であると見なされます。そうでない場合は、変数はピア変数として処理されます。内部変数リストが空の場合は、要求はデータを伴わずに送信されます。これにより、リモート・サーバーはデフォルトの変数表示を戻します。
rl [assocID]	読み取り変数要求を送信して、指定された変数の値がサーバーによって戻されるように要求します。アソシエーション ID が省略されるかまたはゼロが指定される場合は、変数はシステム変数です。そうでない場合は、変数はピア変数であり、戻される値は対応するピアの値になります。変数リストを省略すると、データを伴わない要求が送信されます。これにより、サーバーはデフォルトの変数表示を戻します。 NTPv3 から派生する変数のエンコードおよび意味は、RFC-1305 に記載されています。追加の NTPv4 変数のエンコードおよび意味は、このページの後に示されています。
readvar assocID variable_name [= value] [...]	指定された変数が書き込まれることを除き、 readvar 要求に類似しています。
rv assocID [variable_name [= value] [...]	変数の内部リストが書き込まれることを除き、 readlist 要求に類似しています。
writevar assocID variable_name [= value [...]	
writelist [assocID]	

ファイル

項目	説明
<code>/usr/sbin/ntp4/ntpq4</code>	ntpq コマンドが入っています。 <code>/usr/sbin</code> ディレクトリーから NTP バージョン 4 バイナリーへのデフォルト・シンボリック・リンク。 <code>/usr/sbin/ntpq --> /usr/sbin/ntp3/ntpq</code>

関連資料:

- 250 ページの『[ntpdate4 コマンド](#)』
- 274 ページの『[ntptrace4 コマンド](#)』
- 248 ページの『[ntpdate コマンド](#)』
- 『[ntptrace コマンド](#)』

関連情報:

xntpdc コマンド

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

項目	説明
-d	出力のデバッグをオンにします。
-n	ホスト名ではなくホストの IP アドレスを出力します。
-r Retries	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。
-tTimeOut	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。
-v	詳細モードを指定します。

パラメーター

項目	説明
<i>Server</i>	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。

項目	説明
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

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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, sync distance 0.144135
server2.bozo.com: stratum 2, offset 0.0124263, sync distance 0.115784
usndh.edu: stratum 1, offset 0.0019298, sync 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

項目	説明
<code>/usr/sbin/ntptrace</code>	ntptrace コマンドが入っています。

関連資料:

263 ページの『`ntpq` コマンド』

248 ページの『`ntpdate` コマンド』

関連情報:

`xntpdc` コマンド

`xntpd` コマンド

ntptrace4 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 [ -n ] [ server ]
```

説明

ntptrace コマンドは、Network Time Protocol (NTP) サーバーの時間ソースを決定し、NTP サーバーのチェーンをそれらのマスター時間ソースまでさかのぼって追跡します。引数が指定されない場合は、ローカル・ホストから始まります。**ntptrace** コマンドの出力例を以下に示します。

```
% ntptrace
localhost: stratum 4, offset 0.0019529, sync distance 0.144135
server2ozo.com: stratum 2, offset 0.0124263, sync distance 0.115784
usndh.edu: stratum 1, offset 0.0019298, sync distance 0.011993, refid 'WWVB'
```

各行の左から右に向けての各フィールドは、ホスト名、ホスト階層、および **ntptrace** コマンドで測定したとおりの、そのホストとローカル・ホストの間の時間オフセットです。ローカル・ホスト、ホスト同期間隔、および参照クロック ID (stratum-1 サーバーにのみ適用される) がゼロになるとは限らないのはこのためです。時間はすべて秒数で与えられます。stratum は 1 次ソースに対するサーバー・ホップ・カウントであり、同期間隔は 1 次ソースと比較した推定エラーであることに注意してください。これらの用語は RFC-1305 で正確に定義されています。

フラグ

項目	説明
<code>-n</code>	ホスト名の表示をオフにします。その代わりに、ホスト IP アドレスが表示されます。これは、ネームサーバーが停止している場合に役立つことがあります。

パラメーター

項目	説明
<code>Server</code>	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。

項目	説明
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 サーバーの時間ソースをトレースするには、次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。

```
loopback: stratum 5, offset 0.000076, synch distance 0.18291
ganga08.in.ibm.com: stratum 4, offset -0.001854, synch distance 0.30600
ganga10.in.ibm.com: stratum 3, offset 0.000251, synch distance 0.30550
ausgsa.austin.ibm.com: stratum 2, offset -0.010158, synch distance 0.01921
gsantp.austin.ibm.com: stratum 1, offset 0.016067, synch distance 0.00000, refid
'GPS'
```

ファイル

項目	説明
<code>/usr/sbin/ntp4/ntpdate4</code>	<code>ntpdate</code> コマンドが入っています。
	<code>/usr/sbin</code> ディレクトリーから NTP バージョン 3 バイナリーへのデフォルト・シンボリック・リンク。 <code>/usr/sbin/ntpdate --> /usr/sbin/ntp3/ntpdate</code>

関連資料:

- 250 ページの『`ntpdate4` コマンド』
- 268 ページの『`ntpqq4` デーモン』
- 263 ページの『`ntpqq` コマンド』
- 245 ページの『`ntpd4` デーモン』

関連情報:

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

項目	説明
<i>/usr/sbin/acct</i>	アカウント・コマンドが入っています。
<i>/var/adm/acct/sum</i>	アカウント・データ・ファイルが入っています。

関連資料:

450 ページの『*prdaily* コマンド』

関連情報:

acctmerg コマンド

システム・アカウント

アカウント・サブシステムの設定

コマンドとサブルーチンのモニターとチューニング

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

項目	説明
<code>/usr/games</code>	システム・ゲームが入っています。

関連資料:

615 ページの『`quiz` コマンド』

関連情報:

`arithmetic` コマンド

`back` コマンド

`craps` コマンド

`wump` コマンド

O

以下の AIX コマンドは、文字 *o* から始まります。

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

タイプの文字列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 ] [ -I ]  
[ -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。**-bcCDdFfOoSstvXx** の各オプションを複数使用することで、複数のタイプを指定できます。

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、16 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 や **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。

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。

項目	説明
-A AddressBase	入力オフセット・ベースを指定します。AddressBase 変数には以下の文字の 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 が指定されている場合を除き、出力行の前には、次に書き込まれるバイトの入力オフセット (入力ファイル間で累積されます) があります。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 address_base 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 進数で表示されません。
-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、少なくともスキップ・バイトの長さが無い場合は、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せん。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せん。 ¥ Backslash ¥a Alert ¥b Backspace ¥f Form-feed ¥n New-line character ¥0 Null ¥r Carriage return ¥t Tab ¥v Vertical tab

項目	説明
d	<p>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。</p> <p>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。</p>
f	<p>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。</p> <p>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 タイプの項目に変換を適用するように指定できます。</p>
o	<p>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。</p> <p>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。</p>
u	<p>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。</p> <p>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。</p>
x	<p>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。</p> <p>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。</p>

2 番目のフォーマットのフラグは以下のとおりです。

項目	説明
-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	<p>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ず。</p> <ul style="list-style-type: none"> ¥ Backslash ¥a Alert ¥b Backspace ¥f Form-feed ¥n New-line character ¥0 Null ¥r Carriage return ¥t Tab ¥v Vertical tab <p>その他の文字は 3 桁の 8 進数として表示されます。</p>
-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。

項目	説明
-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。

項目	説明
-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。

項目	説明
-v	すべての入力データを書き出します。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、各出力行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。

項目	説明
0	すべての入力ファイルが正常に処理されました。
>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せん。

ファイル

項目	説明
<code>/usr/bin/od</code>	od コマンドが入っています。

関連資料:

352 ページの『[pg コマンド](#)』

関連情報:

[dbx コマンド](#)

[ナショナル・ランゲージ・サポートの概要](#)

[ロケール環境変数について](#)

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・ファイルで検出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。ファイルが指定しないと、入力 は `stdin` (標準入力) から取得されます。

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 ¥ (円記号) 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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX ユーザー**への注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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
```

html を参照してください。

関連情報:

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、ODM オブジェクト・クラスおよびオブジェクトについて、オブジェクト・クラスの作成方法、ODM オブジェクト・クラスおよびオブジェクトについて、オブジェクト・クラスおよびオブジェクトの保管方法、オブジェクト・クラスの作成方法

odm_add_obj コマンド

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

項目	説明
-o <i>ObjectClass</i>	修正するオブジェクト・クラスを指定します。
-q <i>Criteria</i>	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、 <code>html</code> を参照してください。

関連資料:

283 ページの『`odmadd` コマンド』

関連情報:

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、ODM ディスクリプターについて

`odm_change_obj` コマンド

ODM コマンドとサブルーチンのリスト

ODM のコードおよび出力例

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。

項目	説明
ファイル	: classes
classes	: class classes class
class	: head body tail
head	: struct <i>ClassName</i> {
tail	: }
body	: elements
elements	: elements elements element
element	: char <i>DescriptorName</i> [<i>DescriptorSize</i>];
	vchar <i>DescriptorName</i> [<i>DescriptorSize</i>];
	binary <i>DescriptorName</i> [<i>DescriptorSize</i>];
	short <i>DescriptorName</i> ;
	long <i>DescriptorName</i> ;
	long64 または int64 または ODM_LONG_LONG <i>DescriptorName</i> ;
	method <i>DescriptorName</i> ;
	link <i>StdClassName StdClassName ColName DescriptorName</i> ;

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 ビットのデータベースです。 **long** タイプは、クラス記述ファイルで使用される場合 32 ビット・データ・エンタリーです。 **long64** または **int64** タイプは、クラス記述ファイルで使用される場合 64 ビット・データ・エンタリーです。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

項目	説明
-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	<i>ClassDescriptionFile</i> 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 */
```

関連情報:

オブジェクト・データ・マネージャー (ODM) の概要、ODM オブジェクト・クラスおよびオブジェクトについて、ODM ディスクリプターについて

odm_create_class コマンド

ODM コマンドとサブルーチンのリスト

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

項目	説明
-o <i>ObjectClass</i>	削除元となるオブジェクト・クラスを指定します。
-q <i>Criteria</i>	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、 html を参照してください。

関連情報:

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、ODM オブジェクト・クラスおよびオブジェクトについて

[odm_rm_obj](#) コマンド

[ODM コマンドとサブルーチンのリスト](#)

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

項目	説明
-o <i>ClassName</i>	削除するオブジェクト・クラスを指定します。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、[lssecattr](#) コマンドまたは [getcmdattr](#) サブコマンドの項を参照してください。

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:

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、ODM オブジェクト・クラスおよびオブジェクトについて

odm_rm_class コマンド

ODM コマンドとサブルーチンのリスト

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

項目	説明
-q <i>Criteria</i>	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。

関連資料:

283 ページの『**odmadd** コマンド』

関連情報:

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、ODM オブジェクト・クラスおよびオブジェクトについて

ODM のコードおよび出力例

ODM コマンドとサブルーチンのリスト

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** コマンドまたは『**ODM** のコードおよび出力例』(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) を参照してください。

関連資料:

285 ページの『**odmcreate** コマンド』

関連情報:

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、ODM オブジェクト・クラスおよびオブジェクトについて

ODM のコードおよび出力例

ODM コマンドとサブルーチンのリスト

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

項目	説明
-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

項目	説明
/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。

関連資料:

749 ページの『**rexcd** デーモン』

関連情報:

ネットワーク・ファイルシステム (NFS) の概要

NFS コマンドのリスト

openpts コマンド

目的

リモート・システムの登録および認証を許可します。

構文

```
openpts [-i [-f ] | [-v] | -r | -D] [-h ] [-V] [-u] [-l username] [-p port] [-c configfile] host
```

説明

openpts コマンドを使用すると、システム (ペリファイヤー) がリモート **host** (コレクター) に接続し、そのコレクターがトラステッド・ブートを実行したかどうかを判別することができます。マシンは、コレクターのトラステッド・プラットフォーム・モジュール (TPM) がペリファイヤーによって保守されている基準測定値セット (リファレンス・マニフェスト) に対して整合性をもつと確認された場合に、トラステッド・ブートを実行したとみなされます。基準測定値セットを獲得するために、ペリファイヤーはまず **-i** オプションを使用してコレクターを登録する必要があります。登録の後、ペリファイヤーは保全性レポートに示されている現行値と基準セットとを比較するデフォルトの **-v** オプションを使用して、コレクターを認証することができます。この操作の成功または失敗は、失敗の理由とともにユーザーに報告されます。認証失敗の原因となる操作の例としては、異なるデバイスからのブート、ブート・フラグの変更、ブート・イメージの変更があげられます。

コレクターの状態が更新が処理中になっている場合 (例えば、次のブート操作に影響を与える OS のアップグレード)、それらの更新が認証中に報告されます。ユーザーには、新しい値を受け入れるか拒否するかのプロンプトが出されます。**-u** オプションを使用すれば、更新を自動的に受け入れることができます。認証要求は、コレクターとベリファイヤーの間の通信メカニズムとして、セキュア・シェル (SSH) を使用します。**openpts** コマンドは、ssh コマンド・ユーザー名の **-l** やポートの **-p** などのパラメーターを使用します。

フラグ

項目	説明
-c <i>configfile</i>	使用する構成ファイルを指定します。デフォルトは <code>~/openpts/openpts.conf</code> です。
-D	ターゲットおよびすべてのオプションの構成設定を表示します。
-h	コマンド使用方法の情報を表示します。
-i [-f]	新規のコレクター・パーティションを登録するか、既存のコレクターの登録を強制します。
-l <i>username</i>	ssh コマンド・ユーザー名を指定します。
-p <i>port</i>	ssh コマンド・ポート番号を指定します。
-r	ターゲット・システムに関する情報をすべて除去します。
-u	yes オプションのプロンプトを出さずに、コマンドがコレクターからマニフェストの更新を受け入れられるようにします。デフォルトは no です。
-v (default)	既存のリファレンス・マニフェストに対してコレクターを検査します。
-V	情報を詳細モードで表示します。複数の -V オプションによって詳細度が増します。これはデータのデバッグに使用されます。

ファイル

項目	説明
<code>~/openpts/</code>	このディレクトリーは、すべての構成およびリモート・ホスト情報のデフォルト・ロケーションです。
<code>~/openpts/openpts.conf</code>	ベリファイヤーの構成。
<code>~/openpts/uuid</code>	ベリファイヤーの UUID ファイル。
<code>~/openpts/UUID/ir.xml</code>	リモート・ホストから受け取った最後の健全性レポート。
<code>~/openpts/UUID/newrm_uuid</code>	新しいリファレンス・マニフェストの UUID ファイル (例えば、システム更新後の次のブート操作のもの)。
<code>~/openpts/UUID/policy.conf</code>	リモート・ホストのプロパティーを検査するポリシー。
<code>~/openpts/UUID/rm_uuid</code>	リファレンス・マニフェストの UUID ファイル。
<code>~/openpts/UUID/UUID/rmN.xml</code>	リモート・ホストのリファレンス・マニフェスト。
<code>~/openpts/UUID/target.conf</code>	リモート・ホストの構成。
<code>~/openpts/UUID/vr.properties</code>	健全性レポートから派生したリモート・ホストのプラットフォーム・プロパティー。

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 [ -K keyfile_path_name]{ -o Operation } [ -F] [-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 [ -K keyfile_path_name] -i sysplan { -x sysplan.xml } [ -d] [ -F]
```

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*]

ネットワーク・デーモンの管理用:

OS_install -S | -U

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* パラメーターによって指示されるオブジェクトは、次の 4 つのタイプのいずれかになります。**Client**、**OS_Resource**、**Remote_Resource**、または **Control_Host**。コマンド操作には、クライアント・システムにオペレーション・システムをインストールするためのネットワーク・インストールを使用可能にする **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。この操作により、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。

Remote_Resource オブジェクトに関係する操作では、**ssh-keygen** コマンドを使用して生成される SSH キーを構成する必要があります。SSH キーは、ssh コマンドをローカル・プラットフォームおよびリモート・ソース・サーバー上で実行するのに必要です。HMC 上では、SSH キーが入っているファイル *keyfile_path_name* のデフォルト名は、`/home/hscroot/ssh_keys` です。このファイル名は、**-K** オプションによってオーバーライドすることができます。他のプラットフォームでは、SSH キー・ファイル用のデフォルトのファイル名などありません。他のプラットフォーム上で **-K** オプションが指定されない場合、SSH キー・ファイルの標準パス名が **OS_install** コマンド・プロセスにアクセス可能である必要があります。

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。

HMC または IVM ネットワーク・デーモンは、**OS_install** オブジェクトを変更しなくとも、**S** オプションおよび **U** オプションを使用して開始および停止することができます。

フラグ

項目	説明
-a <i>attr=value</i>	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削除します。
-F	OS_install の割り当て操作時またはシステム計画のインストール時に、必要に応じて、既存のリモート・サーバー・クライアント・システム・オブジェクトのリセットを許可します。
-i <i>sysplan</i>	System Plan モードを指定します。
-K <i>keyfile_path_name</i>	SSH キーが生成されるファイルの絶対パス名を指定します。
-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 <i>Operation</i>	OS_install オブジェクトに対して行う操作を指定します。
-S	OS_install オブジェクトを変更することなく、ネットワーク・デーモンを開始します。

項目

-t *object_type* | *object_name*

-U

-v

-x *sysplan.xml*

説明

-l 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。

OS_install オブジェクトを変更することなく、ネットワーク・デーモンを停止します。

-l 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
define_client [-a <i>attr=value...</i>] { <i>ClientObjectName</i> }	新しいクライアント・オブジェクトを定義します。	<p>ip_addr クライアントの IP アドレス。</p> <p>mac_addr クライアント・システムのネットワーク・インターフェースの MAC アドレス。</p> <p>gateway クライアント・システムの IP ゲートウェイ・アドレス。</p> <p>subnet_mask クライアント・システムの IP サブネット・マスク。</p> <p>lpar クライアントをインストールする LPAR 名 (netboot 操作に必須の属性)。</p> <p>profile クライアントに使用する LPAR プロファイル (netboot 操作に必須の属性)。</p> <p>managed_system LPAR を含んだ管理対象システムの名前 (netboot 操作に必須の属性)。</p> <p>ctrl_host このクライアントの Hardware Control Host オブジェクトの名前 (netboot 操作に必須の属性)。</p>	<p>adapter_speed クライアント・システムのネットワーク・アダプターの速度。</p> <p>adapter_duplex クライアント・システムのネットワーク・アダプターの二重設定。</p> <p>disk_location 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。</p> <p>vlan_tag 仮想ネットワーク通信のネットワーク・インストール中にイーサネット・フレームのタグ付けに使用する、仮想論理エリア・ネットワーク (VLAN) を指定します。有効な値は 0 から 4094 です。</p> <p>vlan_pri 仮想ネットワーク通信のネットワーク・インストール中にイーサネット・フレームのタグ付けに使用する、仮想論理エリア・ネットワーク (VLAN) を指定します。有効な値は 0 から 7 です。</p>

操作	説明	必須属性	オプションの属性
define_resource [-a attr=value...] {ResourceObjectName}	新しい OS_Resource オブジェクトを定義します。	タイプ AIX または VIOS。 version OS バージョン。 location OS_Resource を常駐させようとしている絶対パス。 source インストール・イメージのソース。	configfile インストール構成ファイル。
define_remote_resource [-a attr=value...] {ResourceObjectName}	新しい Remote_Resource オブジェクトを定義します。	サーバー リモート・リソース・サーバーのホスト名。 タイプ AIX または Linux。 remote_identifier リモート・リソース・サーバー上のリソースまたはリソース・セットの名前。	communication_method SSH 通信方式をサポートします。
define_ctrl_host [-a attr=value...] {ControlHostObjectName}	新しい Hardware Control_Host オブジェクトを定義します。	communication_method SSH 通信方式をサポートします。 hostname 制御ホストのホスト名 (ホスト名 localhost は、HMC 制御ホストで OS_install が実行されている場合に指定できます)。 タイプ hmc または ivm。	なし。

操作	説明	必須属性	オプションの属性
allocate [-F][-a attr=value...] {ClientObjectName}	OS_Resource または Remote_Resource 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 OS_install 環境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 OS_Resource または Remote_Resource が既に割り当てられていると、エラーが発生します。	os_resource クライアント・オブジェクトに割り当てる既存の OS_Resource または Remote_Resource オブジェクト。 remote_resource クライアント・オブジェクトに割り当てる既存の Remote_Resource オブジェクト。 install_resource クライアント・オブジェクトに割り当てる既存の OS_Resource または Remote_Resource オブジェクト。	config_file インストール構成ファイル (OS_Resource オブジェクトに適用します)。
netboot {ClientObjectName}	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。
monitor_installation {ClientObjectName}	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。
deallocate {ClientObjectName}	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 OS_Resource または Remote_Resource を割り当て解除します。	なし。	なし。
remove {ObjectName}	オブジェクトを OS_install 環境から除去します。	なし。	なし。

終了状況

項目
0
>0

説明
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. 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. **Remote_Resource** オブジェクトを (HMC に対して **OS_install** のデフォルトの SSH キー・ファイルを使用して) 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_remote_resource -a server=MyNimServer -a type=AIX
-a remote_identifier=NimResGrp1 myRemoteResource
```

4. **Remote_Resource** オブジェクトを (/home/hscroot/id_dsa ファイルにある、先に生成した ssh-keygen キーを使用して) 定義するには、次のように入力します。

```
OS_install -K /home/hscroot/id_dsa -o define_remote_resource -a server=MyNimServer -a type=AIX -a
remote_identifier=NimResGrp1 myRemoteResource
```

5. 例 2 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
```

または

```
OS_install -o allocate -a install_resource=my53resource myclient01
```

6. 例 3 で定義した **Remote_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、既存のクライアント上でリセットを許可するには、以下のようなコマンドを入力します。

```
OS_install -o allocate -F -a remote_resource=myRemoteResource myclient01
```

または

```
OS_install -o allocate -F -a install_resource=myRemoteResource myclient01
```

7. 例 5 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
```

8. クライアント・オブジェクトの **ctrl_host** 属性に対して指定しようとする **Control_Host** 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 **Control_Host** 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。その場合、クライアント・オブジェクトの制御側ホストは、クライアントに対する **netboot** 操作が実行される HMC または IVM でなければなりません。

9. **netboot** 操作を実行するには、次のように入力します。

```
OS_install -o netboot myclient01
```

10. 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
```

11. 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
```

12. 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・ディレクトリーに存在する OS イメージが **remove** 操作によって除去されます。

SSH の構成

- SSH Rivest-Shamir-Adleman (RSA) キーを生成し、それを HMC HOME ディレクトリー内のアクセス可能な `ssh_keys` ファイルに入れるには、次のコマンドを入力します。

```
ssh-keygen -t rsa -f /home/hscroot/ssh_keys
```

- リモート・リソース・サーバー上で、**ssh-keygen** コマンドを使用して生成される `/home/hscroot/ssh_keys.pub` ファイルの内容を、リソース・サーバーの `.ssh/authorized_keys` ファイルに付加またはコピーします。
- リモート HMC 制御ホストのターゲット・クライアント上で `netboot` 操作を実行するために **OS_install** コマンドを使用する場合には、リモート HMC 上で `hscroot` ユーザーとして以下のコマンドを入力することによって、**ssh-keygen** コマンドを使用して生成される `/home/hscroot/ssh_keys.pub` ファイルの内容をリモート HMC `hscroot` ユーザーの `.ssh/authorized_keys2` ファイルに付加します。

```
mkauthkeys -a '<content_of_ssh_keys.pub>'
```

位置

項目	説明
<code>/usr/sbin/OS_install</code>	
<code>/opt/osinstall</code>	OS_install Perl モジュール・ファイルが入っているディレクトリー。

ファイル

項目	説明
<code>/var/osinstall</code>	
<code>/home/hscroot/ssh_keys</code>	OS_install 環境のための構成ファイルを含んだディレクトリー。 HMC 上の SSH キー用のデフォルトのファイル名。

関連情報:

ネットワーク・インストール管理 (NIM) を使用するインストール

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・レベルと Service Pack 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

項目	説明
-l Level	<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。
-f	oslevel 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。
-g Level	<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。
-q	既知のテクノロジー・レベル (-r フラグを使用時) または Service Pack (-s フラグを使用時) の名前をリストします。これらは、 -l または -g フラグを使用して指定できます。
-r	すべてのフラグをテクノロジー・レベルに適用します。
-s	すべてのフラグを Service Pack に適用します。返されるサービス・パック・レベルは、6100-00-01-0748 フォーマットです。ここで、6100 は基本レベル 6.1.0.0、00 はテクノロジー・レベル 0、01 は Service Pack 1、および 0748 はビルド・シーケンス ID を表します。これは、現行レベルに適用できる有効なテクノロジー・レベルおよびサービス・パックを判別するために使用されます。これより低いビルド・シーケンス ID を持つテクノロジー・レベルまたはサービス・パックを適用しようとする場合失敗します。

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。

```
6.1.0.0
```

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す。

```
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

```
-----  
6100-00-02-0750  
6100-00-01-0748  
6100-00-00-0000
```

8. AIX バージョン 6.1 テクノロジー・レベル 0、Service Pack 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 -s -l 6100-00-01-0748
```

9. AIX バージョン 6.1 テクノロジー・レベル 0、Service Pack 1 よりも後の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 -s -g 6100-00-01-0748
```

ファイル

項目	説明
<code>/usr/bin/oslevel</code>	<code>oslevel</code> コマンドが入っています。

関連情報:

lslpp 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・ラインの長さは、200 文字より短くしてください。

ローカル・コマンド

項目	説明
?	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。
?R	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。
d	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 ospf_monitor 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。
h	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。
x	ospf_monitor プログラムを終了します。
@ remote_command	<i>remote_command</i> を同じ (直前の) 宛先に送信します。
@dest_index remote_command	<i>remote_command</i> を構成済み宛先 <i>dest_index</i> に送信します。
F filename	すべての ospf_monitor 出力を <i>filename</i> に送信します。
S	すべての ospf_monitor 出力を <i>stdout</i> に送信します。

リモート・コマンド

項目	説明
a area_id type ls_id adv_rtr	リンク状態通知を表示します。 <i>area_id</i> は、照会の送信先である OSPF 領域です。 <i>adv_rtr</i> 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。 <i>Type</i> 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。 <ol style="list-style-type: none">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、<i>ls_id</i> 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、<i>ls_id</i> 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、<i>ls_id</i> 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、<i>ls_id</i> 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、<i>ls_id</i> 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。

項目	説明
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。
I [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 lsdB 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 lsdB 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。 <ul style="list-style-type: none"> 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 コマンド

p

以下の AIX コマンドは、文字 *p* から始まります。

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。ただし、ラスタ・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** ファイル内の `acctfile =` 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** ファイルが以下のようにになっている場合、`acctfile = /var/adm/lp0acct`

pac コマンドは要約ファイルに `/var/adm/lp0acct_sum` という名前を付けます。

フラグ

項目	説明
-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	アカウント・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・ラインから入力された出力先オプションは、 LPDEST と PRINTER のどちらの環境変数よりも優先されます。
-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
```

このコマンドは印刷されたページ数と料金を、ユーザー名でソートして表示します。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

項目	説明
<code>/usr/sbin/pac</code>	pac コマンドが入っています。
<code>/etc/qconfig</code>	ファイルへのパスを指定します。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

関連情報:

acctcms コマンド

acctcom コマンド

qconfig コマンド

アカウントティング・サブシステムの設定

システム・アカウントティング

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。呼び出し側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、**compress** 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

項目	説明
-f	<i>File</i> 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パックするときは便利です。

パラメーター

項目	説明
ファイル	パックするファイルを指定します。
-	<i>File</i> 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・ラインでさらに - (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のオン/オフ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。

項目	説明
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、変更され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

項目	説明
/usr/bin/pack	pack コマンドが入っています。

関連資料:

340 ページの『**pcat** コマンド』

関連情報:

cat コマンド

unpack コマンド

ファイル・コマンド

入出力ダイレクト

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 フォルダー内のメッセージを、指定された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 文字で区切られます。

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

項目	説明
-file <i>File</i>	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、このフラグ名は略さずに完全なスペルを入力する必要があります。
<i>Messages</i>	圧縮するメッセージを指定します。 <i>Messages</i> 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。
<i>Number</i>	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。
<i>Sequence</i>	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。

項目	説明
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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

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

関連情報:

`inc` コマンド

`.mh_alias` コマンド

`.mh_profile` コマンド

メール・アプリケーション

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

項目	説明
-R <i>module_name</i>	<code>/usr/lib/security/modules.cfg</code> 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) アクセス権限を **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
```

関連資料:

310 ページの『paginit コマンド』

311 ページの『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

項目	説明
<code>/usr/bin/pagesize</code>	<code>pagesize</code> コマンドが入っています。

関連情報:

`getpagesize` コマンド

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

項目	説明
<code>-R <i>module_name</i></code>	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

項目	説明
<i>username</i>	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。 <code>root</code> 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。

関連資料:

308 ページの『`pagdel` コマンド』

『`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

項目	説明
-R <i>module_name</i>	ロード・モジュール <i>module_name</i> 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。

関連資料:

308 ページの『`pagdel` コマンド』

310 ページの『`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** **-a**] [*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は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。

項目	説明
<code>dictionlist</code>	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。
<code>histexpire</code>	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。
<code>histsize</code>	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。
<code>maxage</code>	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。
<code>maxexpired</code>	ユーザーがパスワードを変更できる期間で、 <code>maxage</code> を超える期間の最大値を週単位で指定します。
<code>maxrepeats</code>	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。
<code>minalpha</code>	英字の最小数を指定します。
<code>minother</code>	その他の文字の最小数を指定します。
<code>minlen</code>	文字の最小数を指定します。 注: この値は、 <code>minalpha</code> に <code>minother</code> の値を足したものか、または <code>minlen</code> の値のいずれか大きい方によって決まります。
<code>mindiff</code>	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。 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 <code>abcd</code> で旧パスワードが <code>edcb</code> の場合、異なる文字数は 1 です。
<code>minage</code>	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。
<code>minloweralpha</code>	英小文字の最小数を指定します。
<code>minupperalpha</code>	英大文字の最小数を指定します。
<code>mindigit</code>	数字の最小桁数を指定します。
<code>minspecialchar</code>	特殊文字の最小数を指定します。
<code>pwdchecks</code>	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かまいけません。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。

ユーザーの鍵ストアが存在せず、しかもユーザーの **efs_keystore_access** 属性値が **none** でない場合、**passwd** コマンドはユーザー鍵ストアを作成します。鍵ストアは、`/etc/security/user` ファイルにある暗号化ファイルシステム (EFS) 属性を使用して作成されます。古いパスワードで鍵ストアを開くことができる場合は、鍵ストア・パスワードも変更されます。つまり、ログイン・パスワードと鍵ストア・パスワードが同じ場合、**passwd** コマンドは両方のパスワードを変更します。ファイルシステムが暗号化ファイルシステム (EFS) の場合は、コマンドは **-a** フラグが指定されている場合と同様に機能します。**-a** フラグを指定した場合、パスワード変更後の EFS パスワードはユーザー・ログイン・パスワードと同期されません。したがって、次のログイン時に鍵ストアは自動的にロードされません。

フラグ

項目	説明
-a	すべてのモジュール (compat, LDAP, NIS, など) 内のユーザー・パスワードを変更します。
-f	finger 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、 <code>/etc/passwd</code> ファイルにフルネームを入れることができます。
-s	ログイン・シェルを変更します。
-R load_module	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。root ユーザーとして、`/etc/security/login.cfg` の **usw** スタンザにある **auth_type** 属性の値を **PAM_AUTH** に変更すると、認証に PAM を使用するシステム全体の構成が設定されます。

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` エントリーが必要です。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
```

RBAC ユーザーおよび **Trusted AIX ユーザー**への注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティー」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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 gecost:
"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

項目	説明
/usr/bin/passwd	passwd コマンドが入っています。
/etc/passwd	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。
/etc/security/passwd	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

newpass コマンド

ネットワークの保護

トラステッド・コンピューティング・ベース

shells コマンド

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・ライン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・ファイルでなくてもはなりません、行の長さは無制限です。

フラグ

項目	説明
-d <i>List</i>	出力内で対応する行を区切る区切り文字を、 <i>List</i> 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)。 <i>List</i> 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、 <i>List</i> パラメーターの文字でなく常に改行文字で終了します。 <i>List</i> パラメーター内では以下の特殊文字も使えます。 ¥n 改行文字 ¥t タブ ¥¥ 円記号 ¥0 空文字列 (null 文字ではない) c 拡張文字
-s	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。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。 <i>List</i> 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。

項目	説明
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 列目にとりように入れられます。

これは次のコマンドと等価です。

```
ls | paste -d"%t%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

項目	説明
<code>/usr/bin/paste</code>	<code>paste</code> コマンドが入っています。

関連資料:

446 ページの『`pr` コマンド』

関連情報:

`cat` コマンド

`cut` コマンド

ファイル・コマンド

入出力ダイレクト

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 | -u ] [ -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 -c` または `-u` コマンドで作成される差分に関するリスト (`diff` リスト) が入っており、`diff` コマンド出力の 1 つまたは複数のセットを、通常 *hunk* (ハンク) と呼びます。

`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、

- **diff** のタイプがコピー・コンテキストである場合、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。
 - **diff** のタイプが統合コンテキストである場合、ファイル名は **---** (3 個のダッシュ) または **+++** (3 個のプラス) で始まる行で判別されます。**---** 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**、**-n**、または **-u** フラグを指定すると、**patch** コマンドは各ハンク内の情報を、それぞれコピー・コンテキストの差分、**ed** エディターの差分、通常の差分、または統合コンテキストの差分として、おののを解釈します。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ければ、**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ピー・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・ファイルにはコピー・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、**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どうかは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を仮定します。

```
#ifdef
... NEWCODE
#else
... OLDCODE
#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、正常な終了状況を戻した可能性があります。

フラグ

項目	説明
-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・ファイルをコピー・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 、 -n 、または -u 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。 <pre>#ifdef Define ... (NEWCODE) #else ... (OLDCODE) #endif /* Define */</pre> <p><i>Define</i> 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にのみ機能します。</p>
-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 、 -n 、または -u 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に、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 、 -e 、または -u 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。
-p Number	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ー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ー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、 <ul style="list-style-type: none">• -p 0 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。• -p 1 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。• -p 4 を指定すると、4 つのスラッシュと 3 つのディレクトリーが除去され、<code>blurf1/blurf1.c</code> となります。 <p>-p 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>
-r RejectFile	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 .rej を付加して作成されます。

項目	説明
-R	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、 -R フラグを指定すると、 patch 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。 -R フラグは ed 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。 -R フラグを指定しないと、 patch 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、 -R 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。 注: 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。
-s	エラーが発生しない限り、何も表示せずにパッチします。
-u	統一コンテキストの差分としてパッチ・ファイルを解釈します (-u または -U フラグを指定時の diff コマンドの出力)。このフラグは、 -c 、 -e 、または -n フラグと一緒に指定できません。
-v	改訂ヘッダーとパッチ・レベルを出力します。 -v 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。
-x Number	内部デバッグ・フラグを設定します。このフラグは patch コマンド開発者専用です。

終了状況

次の終了値が戻されます。

項目	説明
0	正常終了。
1	エラーが発生しました。

例

1. **diff**listing 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。

3. 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

項目	説明
<code>/usr/bin/patch</code>	<code>patch</code> 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 ] [ -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、`pathname` 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。

以下のいずれかの条件が満たされる場合は、エラーではありません。1 つは、パス名の 1 つ以上のコンポーネントが存在しない場合です。もう 1 つは、`pathname` パラメーターで指定されたパス名と一致するファイルを作成でき、そのファイルが上記のどの条件にも違反してはならない場合です。

`-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

項目	説明
<code>-p</code>	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。 <ul style="list-style-type: none"> • 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。
<code>-P</code>	<code>pathname</code> オペランドをチェックし、 <code>pathname</code> オペランドが以下の基準を満たしている場合はエラー・メッセージを戻します。 <ul style="list-style-type: none"> • <code>pathname</code> オペランドに、先頭文字がハイフンである構成要素が含まれている。 • <code>pathname</code> オペランドが空。

終了状況

このコマンドは、以下の終了値を戻します。

項目	説明
0	すべての <i>pathname</i> 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

1. 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
```

2. 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

項目	説明
<code>/usr/bin/pathchk</code>	pathchk コマンドが入っています。

関連情報:

`mhpath` コマンド

ファイルシステム

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 | -n ] [-d] [-U] [ -v ] [ -H | -L ] [-f Archive] [ -s ReplacementString... ] [-x Format]
[-o Options] [ -Z ] [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 | -n ] [ -d ] [ -i ] [ -k ] [ -U ] [ -u ] [ -v ] [ -H | -L ] [ -f Archive] [
-o Options] [ -p String... ] [ -s ReplacementString... ] [ -x Format] [ -Z ] [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] [ -Z ] [ 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] [ -Z ] [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。NULL が入っている区域で、調整済みでサイズ指定された区域をファイルがブロックしている場合、**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が割り当てられることはありません。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、標準入力から読み取られたアーカイブ・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** フラグも指定されない場合は、**-c**、**-d**、**-f**、**-n**、**-s**、および **-v** フラグと *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。

-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。

コピーされたファイルは、アーカイブ・ファイルに書き込まれ、その後で取り出された内容と同じですが、元のファイルとコピーされたファイルの間にハード・リンクがある点が異なります。

-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

読み取りモードまたはコピー・モードでは、アーカイブ・メンバーを取り出すために中間ディレクトリーが必要である場合、**pax** コマンドは **mkdir(0)** サブルーチンに類似したアクションを実行します。この場合、パス引数として中間ディレクトリーが使用され、モード引数として値 **S_IRWXU** が使用されます。

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** コマンドは、最後に訪問されたファイルの祖先である、既に訪問されているディレクトリーを入力することによって、無限ループを検出します。無限ループを検出すると、**pax** コマンドは診断メッセージを標準エラーに書き込み、終了します。

pax コマンドが読み取りモードまたはリスト・モードのときは、**-x pax** アーカイブ・フォーマットを使用することにより、拡張ヘッダー・レコード内のファイル名、リンク名、所有者名、その他のフィールドを **pax UTF8** コード・セット形式から現行コード・セットとロケールに変換できません。**pax** コマンドは、診断メッセージを標準エラーに書き込み、**-o invalid=** オプションで記述されている通りに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 を保持するのに失敗すると、診断メッセージが標準エラーに書き込まれますが、取り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す。**pax** アーカイブ・ファイル内に、拡張属性用エントリーに新しいレコード・タイプが必要です。

変数

項目	説明
<i>Directory</i> ファイル	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。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。 <i>File</i> 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。
<i>Pattern</i>	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。 ¥ (円記号) は、 <i>Pattern</i> 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。 <i>Pattern</i> 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。
	<i>Pattern</i> 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

項目	説明
-a	アーカイブの最後にファイルを追加します。
-b Blocking	注: ストリーミング・テープ・デバイスでは、追加機能は許可されません。 出力用のブロックのサイズを指定します。 <i>Blocking</i> 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。)
	<i>Blocking</i> パラメーターは、以下のいずれかの値を受け入れます。
	Integer b バイト単位のブロック・サイズが、 <i>Integer</i> 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、 <i>Integer</i> 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、 <i>Integer</i> 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、 <i>Integer</i> 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	<i>Pattern</i> パラメーターで指定されたファイルを除く、すべてのファイル・メンバーまたはアーカイブ・メンバーを突き合わせます。
-d	コピー、アーカイブ、または取り出しが行われるディレクトリーが、ディレクトリーの内容ではなく、ディレクトリーに一致するようにします。
-E	新規のアーカイブまたは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・ライン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。

項目	説明
-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。 <i>Pattern</i> 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 <i>File</i> 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・ラインに指定されたか、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 <i>Pattern</i> 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。

項目
-oOptions

説明

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* の中には、リテラル・コンマの前には円記号 (¥) 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。複数の **-o** オプションに指定したキーワードが矛盾する場合は、コマンド・ライン・シーケンス内で後の方に出てくるキーワードと値が優先されます。それより前の値は無視されます。

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なくてはなりません。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される必要があります、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** が含まれている必要があります。

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 (バイト単位) を 10 進数で示します。**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、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。

exthdr.name=string (-x pax フォーマットのみ適用されます。)

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・セットと現行ロケール内に書き込むことができない。

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・セット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・セット形式から現行コード・セット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、ファイルが、アーカイブに内容が書き込まれているファイルへのハード・リンクであっても、**pax** コマンドは、そのファイルの内容をアーカイブに書き込みます。

項目
-o Options (続き)

説明

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** フラグ以降の唯一または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して見なされます。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・ライン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、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** フォーマットのものに適用されます。)

-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ているキーワードと値は、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

文字セットの名前は、後続のファイルのデータをエンコードするために設定されます。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。

項目 値	説明 標準
"ISO-IR 646 1990"	ISO/IEC 646 IRV
"ISO-IR 8859 1 1987"	ISO 8859-1
"ISO-IR 8859 2 1987"	ISO 8859-2
"ISO-IR 10646 1993"	ISO/IEC 10646
"ISO-IR 10646 1993 UTF8"	ISO/IEC 10646, UTF8 encoding
"BINARY"	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

前にアーカイブされた別のファイル (タイプを問わず) に対して作成されるリンクのパス名。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。

書き込みモードまたはコピー・モードで使用される場合、**pax** は、パス名全体を NULL 以外のポータブル文字集合のメンバーで表すことができない各リンクのリンク・パス拡張ヘッダー・レコードを組み込みます。

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

`real time` 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せん。**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 セキュリティー標準化のために予約されません。**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・ライン・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す。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。

項目

-p String

説明

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。

項目

-s ReplacementString

説明

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 / (円記号) 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。

項目
-x *Format*

説明

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 から 32256 まで、512 刻みで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 から 32256 まで、512 刻みで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 から 32256 まで、512 刻みでブロック化値がサポートされます。

- **Filename: pax** コマンドは、システムによって定義された **PATH_MAX** 制限に達するまでのパスとファイル名の長さをサポートします。パスとファイル名の入力の長さが **PATH_MAX** 制限を超えると、値はアーカイブされません。
- **gid** または **uid: pax** コマンドは、**UINMAX** 制限に達するまでの **gid** と **uid** の値をサポートします。**UINMAX** 制限より大きな値は切り捨てられます。

既存のアーカイブ・フォーマットとは異なるフォーマットでアーカイブ・ファイルを追加しようとする、**pax** コマンドは、ゼロ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へ降りることはありません。

-Z 暗号化ファイルまたはディレクトリーの暗号化ファイルシステム (EFS) 情報をアーカイブします。EFS 情報はデフォルトで抽出されます。アーカイブのメンバーがリストされると、**-Z** フラグを指定してアーカイブされた暗号化ファイルおよびディレクトリーについて、ファイル・モードの後に **-e** 標識が表示され、それ以外のファイルについてはハイフン (-) が表示されます。

注: **-Z** フラグで作成されたアーカイブは、AIX 6.1 またはそれ以降でのみ復元できます。

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(0)** 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。
 - 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。

項目	説明
0	正常終了。
>0	エラーが発生しました。

セキュリティ

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

1. **olddir** ディレクトリー階層を **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. 現行ディレクトリーの内容をテープ・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、以下のいずれかのコマンドを入力します。
 - `pax -wvf/dev/rfd0 -s /xxx/XXX/p xxx`
 - `pax -wvf/dev/rfd0 -s/x/X/gp xxx`
4. 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_
```
5. アーカイブ **pax.ar** 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
```
6. **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
```
7. **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
```
8. 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
```
9. 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
```
10. **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
```
11. 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
```
12. **olddir** ディレクトリー階層を、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

項目	説明
<code>/usr/bin/pax</code>	<code>pax</code> コマンドが入っています。

関連資料:

860 ページの『`rmt` コマンド』

関連情報:

`ed` コマンド

`cpio` コマンド

`tar` コマンド

ファイル・コマンド

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。

項目	説明
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

項目	説明
<code>/usr/bin/pcat</code>	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

`unpack` コマンド

ファイル・コマンド

入出力ダイレクト

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。

- `/dev/tty1` デバイスなど、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ません。

フラグ

項目	説明
-a	すべてのポートを遅延状態として使用可能にします。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
/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。

関連資料:

- 『pdisable コマンド』
- 545 ページの『pshare コマンド』
- 552 ページの『pstart コマンド』

関連情報:

- getty コマンド
- init 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ければなりません。

フラグ

項目	説明
-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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**Issecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
/etc/locks	pshare および delay コマンドの lock ファイルが入っています。
/usr/sbin/pdisable	pdisable コマンドが入っています。

関連資料:

- 341 ページの『**pdelay** コマンド』
- 350 ページの『**penable** コマンド』
- 545 ページの『**pshare** コマンド』

関連情報:

init コマンド
inittab コマンド

pdlink コマンド

目的

区画サブディレクトリー内のファイルへのリンクを作成します。

構文

pdlink *dirname filename ...*

説明

pdlink により、区画サブディレクトリの下に存在するファイルが、異なる SL で実行されるプロセスからアクセスできるようになります。ファイルは、呼び出しプロセスの機密ラベル (SL) に対応します。
dirname パラメーターで指定するディレクトリ名は、区画ディレクトリでなければなりません。また、*filename* パラメーターで指定するファイル名は、指定されたディレクトリの下にあるファイル名 (パス名でない) でなければなりません。複数のファイル名を指定できます。

pdlink コマンドは、指定されたファイルへのハード・リンクを作成しますが、以下の制限があります。

- リンクは区画サブディレクトリ内にのみ作成されます。
- 区画サブディレクトリのそれぞれは、**pdlink** コマンドの実行時に存在しなければなりません。
- リンクが作成される区画サブディレクトリは、*filename* パラメーターで指定されたファイルの最小 SL より高い SL を持っていなければなりません。

セキュリティ

権限のあるユーザーだけが **pdlink** コマンドを実行できます。

項目	説明
aix.mls.pdir.link	区画サブディレクトリ内にこのコマンドでリンクを作成するために必要です。

終了状況

pdlink コマンドは、以下の終了値を戻します。

項目	説明
0	正常終了。
>0	エラーが発生しました。

例

1. **partdir** と呼ばれる区画ディレクトリ内にある **sample.c** ファイルのリンクを作成するには、次のように入力します。

```
pdlink partdir sample.c
```

ファイル

項目	説明
/usr/sbin/pdlink	pdlink コマンドが入っています。

関連資料:

345 ページの『**pdmkdir** コマンド』

348 ページの『**pdrmdir** コマンド』

349 ページの『**pdset** コマンド』

346 ページの『**pdmode** コマンド』

関連情報:

Trusted AIX

pdmkdir コマンド

目的

区画ディレクトリーを作成します。

構文

```
pdmkdir [ -m Mode ] [ -u Owner ] [ -g Group ] dirname ...
```

説明

pdmkdir コマンドは、*dirname* パラメーターで指定された区画ディレクトリーを作成します。通常のユーザーは、任意アクセス制御 (DAC)、必須アクセス制御 (MAC)、および必須保全性制御 (MIC) アクセス権によって新規ディレクトリーの作成を許可されている場合に、区画ディレクトリーを作成できます。

aix.mls.pdir.mkdir 権限のあるユーザーは、DAC、MAC、および MIC アクセス権をオーバーライドできます。

フラグ

項目	説明
-g <i>Group</i>	新規作成ディレクトリーのグループを設定します。グループ名またはグループ ID のいずれかを指定できます。 aix.mls.pdir.mkdir 権限のあるユーザーは、ディレクトリーのグループを、自分がメンバーでないグループに変更できます。
-m <i>Mode</i>	新規作成ディレクトリーのアクセス権ビットを、 <i>Mode</i> 変数で指定された値に設定します。 <i>Mode</i> 変数は数値として指定します。
-u <i>Owner</i>	新規作成ディレクトリーの所有者を設定します。所有者名または所有者 ID のいずれかを指定できます。 aix.mls.pdir.mkdir 権限のあるユーザーは、ディレクトリーの所有者を変更できます。

注: 設定された *Mode*、*Owner*、または *Group* 変数は区画ディレクトリー、およびこのコマンドを実行したプロセスの機密ラベル (SL) に基づいて作成された、区画サブディレクトリーに適用されます。異なる SL を持つ別のプロセスが区画ディレクトリーにアクセスした場合、作成された区画サブディレクトリーはこれらのフラグでは制御できません。

セキュリティ

すべてのユーザーが **pdmkdir** コマンドを実行できます。特定の機能を正常に実行するためには、ユーザーは以下の権限を持っていないければなりません。

項目	説明
aix.mls.pdir.mkdir	-u または -g フラグを使用して所有者またはグループを変更するために必要です。この権限は、親ディレクトリーの DAC、MAC、および MIC アクセス権を無視するパスにディレクトリーを作成するためにも必要です。

終了状況

pdmkdir コマンドは、以下の終了値を戻します。

項目	説明
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。

例

1. 区画ディレクトリーを作成するには、次のように入力します。

```
pdmkdir partdir
```

2. アクセス権「755」、ユーザー「joe」、グループ「staff」を指定して区画ディレクトリーを作成するには、次のように入力します。

```
pdmkdir -m 755 -u joe -g staff partdir
```

ファイル

項目	説明
/usr/sbin/pdmkdir	pdmkdir コマンドが入っています。

関連資料:

348 ページの『**pdrmdir** コマンド』

349 ページの『**pdset** コマンド』

『**pdmode** コマンド』

343 ページの『**pdlink** コマンド』

関連情報:

Trusted AIX

pdmode コマンド

目的

仮想または実際の区画ディレクトリー・アクセス・モードでコマンドを呼び出します。

構文

```
pdmode [ [ -r ] command [ arg ... ] ]
```

説明

pdmode コマンドにより、仮想または実際の区画ディレクトリー・アクセス・モードで *command* パラメーターを使用して指定するコマンドを呼び出すことができます。引数を指定しないで呼び出すと、**pdmode** コマンドは、このコマンドを呼び出したプロセスの区画ディレクトリー・アクセス・モードを戻します。

フラグのない **command** パラメーターを続けた *pdmode* コマンドを実行する場合、コマンドは仮想モードで実行されます。ユーザーは、**-r** フラグを使用して、実際の区画ディレクトリー・アクセス・モードでコマンドを実行できます。

フラグ

項目	説明
<code>-r command [arg...]</code>	新規プロセスの区画ディレクトリー・アクセス・モードを実モードに設定します。このモードでは、区画ディレクトリーは透過ではないため、ユーザーは区画ディレクトリーのサブツリーをナビゲートするときに区画ディレクトリーを意識する必要があります。 このオプションを指定してコマンドを正常に実行するためには、ユーザーは aix.mls.pdir.mode 権限を持っている必要があります。

セキュリティ

すべてのユーザーが **pdmode** コマンドを実行できます。特定の機能を正常に実行するためには、以下の権限が必要です。

項目	説明
aix.mls.pdir.mode	<code>-r</code> フラグを指定して pdmode コマンドを実行するために必要です。

終了状況

pdmode コマンドは、以下の終了値を戻します。

項目	説明
0	正常終了。
>0	エラーが発生しました。

例

1. 区画ディレクトリー・アクセス・モードを取得するには、次のように入力します。

```
pdmode
```

2. 仮想モードで **ls** コマンドを実行するには、次のように入力します。

```
pdmode ls -l
```

3. 実モードで **ls** コマンドを実行するには、次のように入力します。

```
pdmode -r ls -l
```

ファイル

項目	説明
<code>/usr/sbin/pdmode</code>	pdmode コマンドが入っています。

関連資料:

- 345 ページの『**pdmkdir** コマンド』
- 348 ページの『**pdrmdir** コマンド』
- 349 ページの『**pdset** コマンド』
- 343 ページの『**pdlink** コマンド』

関連情報:

Trusted AIX

pdrmdir コマンド

目的

区画ディレクトリーを削除します。

構文

pdrmdir *dirname* ...

説明

pdrmdir コマンドは、*dirname* パラメーターで指定された区画ディレクトリーを削除します。通常のコマンドは、任意アクセス制御 (DAC)、必須アクセス制御 (MAC)、および必須保全性制御 (MIC) アクセス権によってディレクトリーの削除を許可されている場合に、区画ディレクトリーを削除できます。

aix.mls.pdir.rmdir 権限のあるユーザーは、DAC、MAC、および MIC アクセス権をオーバーライドできます。

pdrmdir コマンドは、空の区画サブディレクトリーのみを除去し、区画サブディレクトリー内のファイルまたはディレクトリーは除去しません。区画ディレクトリーは、すべての区画サブディレクトリーが除去されてディレクトリーが空になった後で除去されます。ファイルが存在する場合は、区画ディレクトリーの除去は失敗します。

セキュリティ

すべてのユーザーが **pdrmdir** コマンドを実行できます。特定の機能を正常に実行するためには、ユーザーは以下の権限を持っていない限りなりません。

項目	説明
aix.mls.pdir.rmdir	DAC、MAC、および MIC アクセス権を無視してパス内のディレクトリーを除去するために必要です。

終了状況

pdrmdir コマンドは、以下の終了値を返します。

項目	説明
0	正常終了。
>0	エラーが発生しました。

例

1. 区画ディレクトリーを削除するには、次のように入力します。

```
pdrmdir partdir
```

ファイル

項目	説明
<code>/usr/sbin/pdrmdir</code>	<code>pdrmdir</code> コマンドが入っています。

関連資料:

- 345 ページの『`pdmkdir` コマンド』
- 『`pdset` コマンド』
- 346 ページの『`pdmode` コマンド』
- 343 ページの『`pdlink` コマンド』

関連情報:

Trusted AIX

pdset コマンド

目的

通常のディレクトリーを区画ディレクトリーに変換します。

構文

`pdset dirname ...`

説明

`pdset` コマンドは、`dirname` パラメーターで指定された通常のディレクトリーを区画ディレクトリーに変換します。

指定するディレクトリー名は、区画サブディレクトリーの名前または区画サブディレクトリーのサブディレクトリーの名前であってはなりません。このディレクトリーの下に存在するサブディレクトリーまたはファイルは、区画ディレクトリーの実モードでのみアクセスできます。

セキュリティ

権限のあるユーザーだけが `pdset` コマンドを実行できます。

項目	説明
<code>aix.mls.pdir.set</code>	通常のディレクトリーを区画ディレクトリーに変換するために必要です。

終了状況

`pdset` コマンドは、以下の終了値を戻します。

項目	説明
0	正常終了。
>0	エラーが発生しました。

例

- ディレクトリーを区画ディレクトリーに変換するには、次のように入力します。

```
pdset testdir
```

ファイル

項目	説明
<code>/usr/sbin/pdset</code>	<code>pdset</code> コマンドが入っています。

関連資料:

345 ページの『`pdmkdir` コマンド』

348 ページの『`pdrmdir` コマンド』

346 ページの『`pdmode` コマンド』

343 ページの『`pdlink` コマンド』

関連情報:

Trusted AIX

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。

- `/dev/tty1` デバイスなど、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

項目	説明
-a	すべての通常ポートを使用可能にします。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
/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。

関連資料:

- 341 ページの『**pdelay** コマンド』
- 342 ページの『**pdisable** コマンド』
- 355 ページの『**phold** コマンド』

関連情報:

- init** コマンド
- inittab** コマンド

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

項目	説明
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/perfbw

ファイル

項目	説明
/usr/bin/perfbw	perfbw コマンドが入っています。
\$HOME/workspace	設定が入っている perfbw 作業ディレクトリーが入っています。

関連情報:

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 ] [ +/Pattern/ ] [ 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。

項目	説明
<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。
+Number	<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。
-Number	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。
l	(L の小文字) デイスプレィを下方に 1 行スクロールします。
Numberl	<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。
+Numberl	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。
-Numberl	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。
d	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、 d サブコマンドと同様に機能します。
-d	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、 -d サブコマンドと同様に機能します。
Ctrl-L	現在のページを再表示します。1 つの . (ドット) は、 Ctrl-L キー・シーケンス・サブコマンドと同様に機能します。
\$	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。

項目	説明
[<i>Number</i>]/ <i>Pattern</i> /	<i>Number</i> 変数で指定した <i>Pattern</i> 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。 <i>Number</i> 変数のデフォルトは 1 です。
<i>Number</i> ? <i>Pattern</i> ?	

項目	説明
<i>Number</i> ^ <i>Pattern</i> ^	<i>Number</i> 変数で指定した <i>Pattern</i> 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。 <i>Number</i> 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。

項目	説明
[<i>Number</i>] n	<i>Number</i> 変数で指定した数だけ、コマンド・ライン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。
[<i>Number</i>] p	<i>Number</i> 変数で指定した数だけ、コマンド・ライン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。
[<i>Number</i>] w	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、[<i>Number</i>] z サブコマンドと同じです。
[<i>Number</i>] z	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、[<i>Number</i>] w サブコマンドと同じです。
s <i>File</i>	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。 -n 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。
h	使用可能なサブコマンドを省略した要約を表示します。
q または Q	pg コマンドを終了します。
! <i>Command</i>	指定したコマンドを SHELL 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。 -n 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:

1. 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。
2. 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

項目	説明
-c	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、 terminfo ファイルのワークステーション・タイプに、 clear_screen フィールドが定義されていない場合です。
-e	各ファイルの終わりで一時停止しません。
-f	行を分割しません。通常、 pg コマンドは画面幅よりも長い行を分割します。
-n	pg 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。
-p <i>String</i>	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 %d 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、 : (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。
-r	!" 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。
-s	すべてのメッセージとプロンプトを強調表示します。
+LineNumber	指定した行番号から始めます。
-Number	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。
+IPatternI	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。

項目	説明
0	正常終了。
>0	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

項目	説明
<code>/usr/bin/pg</code>	pg コマンドが入っています。
<code>/usr/share/lib/terminfo/*</code>	端末タイプを定義する terminfo ファイルが入っています。
<code>/tmp/pg*</code>	pg 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

入出力ダイレクト

`shells` コマンド

ファイル・コマンド

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

項目	説明
-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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**Issecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
/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。

関連資料:

- 341 ページの『**pdelay** コマンド』
- 342 ページの『**pdisable** コマンド』
- 545 ページの『**pshare** コマンド』

関連情報:

init コマンド
inittab コマンド

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 ] [ - | 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。

項目 説明
ファイル 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。
ル

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。

項目	説明
<i>OptionalWidth</i>	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。
<i>OptionalHeight</i>	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。
<i>ElementList</i>	以下のようなエレメントのリストを表示します。 <i>Shape AttributeList</i> <i>For Statement</i> <i>Placename: Element</i> <i>If Statement</i> <i>Placename: Position</i> <i>Copy Statement</i> <i>Variable = Expression</i> <i>Print Statement</i> <i>Direction</i> <i>Plot Statement</i> { <i>List of Elements</i> } <i>sh X Commandline X</i> [<i>List of Elements</i>] <i>troff-command</i>

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 ¥ (円記号) 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
h(eigh)t <i>Expression</i>	wid(th) <i>Expression</i>
rad(ius) <i>Expression</i>	diam(eter) <i>Expression</i>
up <i>OptionalExpression</i>	down <i>OptionalExpression</i>
right <i>OptionalExpression</i>	left <i>OptionalExpression</i>
from <i>Position</i>	to <i>Position</i>
at <i>Position</i>	with <i>Corner</i>
by <i>Expression, Expression</i>	then
dotted <i>OptionalExpression</i>	dashed <i>OptionalExpression</i>
chop <i>OptionalExpression</i>	-> <- <->
invis	same
<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。

項目	説明
図形エレメント	属性
box	h(eigh)t , wid(th) , at , same , dotted , dashed , invis , <i>Text</i>
circle , ellipse	rad(ius) , diam(eter) , h(eigh)t , wid(th) , at , same , invis , <i>Text</i>
arc	up , down , left , right , h(eigh)t , wid(th) , from , to , at , rad(ius) , invis , <i>ccw</i> , <i>cw</i> , <i><-</i> , <i>-></i> , <i><-></i> , <i>Text</i>
line , arrow	up , down , left , right , h(eigh)t , wid(th) , from , to , by , then , at , same , dotted , dashed , invis , <i><-</i> , <i>-></i> , <i><-></i> , <i>Text</i>
spline	up , down , left , right , h(eigh)t , wid(th) , from , to , by , then , at , same , invis , <i><-</i> , <i>-></i> , <i><-></i> , <i>Text</i>
move	up , down , left , right , to , by , same , <i>Text</i>
<i>Text-list</i>	at , <i>Text-item</i>

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・エンタリーがあると、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タリーに独自に適用されます。

テキスト・エンタリーには、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。

項目
位置/ロケーション

説明
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。

項目	説明
boxwid	0.75
boxht	0.5
circlerad	0.25
arcrad	0.25
ellipsewid	0.75
ellipseht	0.5
linewid	0.5
lineht	0.5
movewid	0.5
moveht	0.5
arrowwid	0.05
arrowht	0.1
textwid	0
textht	0
dashwid	0.5
scale	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。

項目	説明
!	否定 (ではない)
>	よりも大きい
<	よりも小さい
>=	よりも大きいか等しい
<=	よりも小さいか等しい
&&	And
	または
==	等しい
!=	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・ラインを実行します。

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

項目	説明
-TName	指定された印刷装置の出力を作成します。 <i>Name</i> 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 **-T Name** 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

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 |
-nopublic ] [ -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

項目	説明												
-after Date	<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 border="0"> <tbody> <tr> <td>yesterday</td> <td>today</td> <td>tomorrow</td> </tr> <tr> <td>sunday</td> <td>monday</td> <td>tuesday</td> </tr> <tr> <td>wednesday</td> <td>thursday</td> <td>friday</td> </tr> <tr> <td>saturday</td> <td>-Days</td> <td>SystemDate</td> </tr> </tbody> </table>	yesterday	today	tomorrow	sunday	monday	tuesday	wednesday	thursday	friday	saturday	-Days	SystemDate
yesterday	today	tomorrow											
sunday	monday	tuesday											
wednesday	thursday	friday											
saturday	-Days	SystemDate											
-and	pick コマンドは、曜日を過去の日と見なします。例えば、 monday は、その当日、または次週の月曜日ではなく、直前の月曜日を意味します。 -Days 引数を使用して、経過した日数を指定することもできます。例えば、 -31 は、31 日前を意味します。 SystemDate 引数の場合、システムに定義した有効なフォーマットを指定できます。 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、 pick -after Sunday -and -from mark のようにします。 -and フラグは、 -or フラグより優先順位が高く、 -not フラグよりも優先順位が低くなります。 -lbrace および -rbrace 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。												
-before Date	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、 -after フラグを参照してください。												
-cc "Pattern"	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。												
-date "Pattern"	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。												
-datefield Field	-after および -before 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、 pick コマンドは Date: フィールドを使用します。												
+Folder	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。												
-from "Pattern"	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。												
-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。												
-lbrace	-and 、 -or 、および -not 演算をグループ化します。 -lbrace と -rbrace フラグの間の演算は、1 つの演算として実行されます。 -lbrace および -rbrace フラグをネストすることができます。												

項目	説明
-list	<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、pick 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>
Messages	<p>シーケンスを指定しない場合、デフォルトは、-list 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> <p>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。</p> <p>cur または . (ピリオド) 現行メッセージ。</p> <p>first フォルダー内の最初のメッセージ。</p> <p>last フォルダー内の最後のメッセージ。</p> <p>new 新たに作成したメッセージ。</p> <p>next 現行メッセージの次のメッセージ。</p> <p>prev 現行メッセージの前のメッセージ。</p>
-nolist	pick 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、 -nolist フラグです。
-nopublic	シーケンスの使用に制限を与えます。 -nopublic 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。
-not	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、 pick -not -from george 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。 -not フラグは、 -and フラグより優先順位が高く、 -and フラグは、 -or フラグより優先順位が高くなります。 -lbrace および -rbrace 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。
-nozero	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。
-or	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、 pick -from amy -or -from mark などです。 -not フラグは、 -and フラグより優先順位が高く、 -and フラグは、 -or フラグより優先順位が高くなります。 -lbrace および -rbrace 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。
-public	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。 -public 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。
-rbrace	-and 、 -or 、および -not 演算をグループ化します。 -lbrace と -rbrace フラグの間の演算は、1 つの演算として実行されます。 -lbrace および -rbrace フラグをネストすることができます。
-search "Pattern"	メッセージの任意の位置にある "Pattern" 変数で指定した文字列を含むメッセージを選択します。
-sequence Name	Name 変数で指定したシーケンスで pick コマンドを指定して、選択したメッセージを格納します。
-to "Pattern"	To: フィールドの "Pattern" 変数で指定した文字列を含むメッセージを選択します。
-zero	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。
—Component "Pattern"	Component 変数で指定したヘッダー・フィールドの "Pattern" 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。

プロファイル・エントリー

次のプロファイル・エントリーは、`UserMHDDirectory/.mh_profile` ファイルの一部です。

項目	説明
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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
<code>\$HOME/.mh_profile</code>	ユーザーの MH プロファイルが入っています。
<code>/usr/bin/pick</code>	pick コマンドが入っています。

関連情報:

`mark` コマンド

`.mh_alias` コマンド

メール・アプリケーション

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

項目	説明
<code>-c Count</code>	<code>Count</code> 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。
<code>-w timeout</code>	このオプションが作用するのは、 <code>-c</code> 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、 <code>ping</code> 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。
<code>-d</code>	ソケット・レベルのデバッグを開始します。
<code>-D</code>	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。
<code>-f</code>	<code>flood-ping</code> オプションを指定します。 <code>-f</code> フラグは、パケットが戻るのと同じ速さ、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。 <code>flood</code> で <code>ping</code> するには、root ユーザーしか使用できません。 <code>-f</code> フラグと <code>-i Wait</code> フラグには互換性がありません。
<code>-I a.b.c.d</code>	<code>a.b.c.d</code> 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。 <code>-I</code> フラグは、大文字の <code>i</code> です。
<code>-o interface</code>	<code>interface</code> 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。

項目	説明
-i <i>Wait</i>	各パケットの送信の間では <i>Wait</i> 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 <i>Preload</i>	<i>Preload</i> 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 1 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 <i>Pattern</i>	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。 <i>Host</i> 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 できません。
-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タを表示します。 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 <i>addr_family</i>	<i>addr_family</i> 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 <i>PacketSize</i>	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 <i>hostname/IP addr</i>	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 <i>ttl</i>	マルチキャスト・パケットの生存時間が <i>ttl</i> 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

項目	説明
<i>PacketSize</i>	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。
<i>Count</i>	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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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) を受信するまで継続します。

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。

関連資料:

44 ページの『netstat コマンド』

関連情報:

ifconfig コマンド

通信およびネットワーク

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

項目	説明
-a <i>Attribute</i>	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 <i>QueueDeviceName</i>	仮想プリンター定義の <i>QueueDeviceName</i> 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 <i>Action</i>	<p>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。</p> <p>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。</p> <p>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。</p> <p>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す。</p> <p>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。</p> <p>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。</p> <p>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。</p>
-q <i>PrintQueueName</i>	仮想プリンター定義の <i>PrintQueueName</i> 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
```

- 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

項目	説明
<code>/usr/sbin/piottred</code>	<code>piottred</code> コマンドが入っています。

関連情報:

`lsvirprt` コマンド

仮想プリンターの定義と属性

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ーパー・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 (このトピック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・ラインに直接 `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

項目

-a PreviewOption

説明

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・ライン・フラグ値です。これらの値は、コマンド・ライン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。

項目

-d *InputDataType*

説明

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。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、html で使用できるフィルター・フラグに類似しています。

-f *FilterType*

関連資料:

446 ページの『pr コマンド』

関連情報:

lpr コマンド

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

キューを追加しない場合のプリンターの構成

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 **pio** 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。

項目	説明
%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。

項目	説明
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

項目	説明
-H <i>HostName</i>	<i>HostName</i> 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

項目	説明
<code>/usr/lpd/pio/etc/pioburst</code>	pioburst コマンドが入っています。

関連資料:

372 ページの『**piobe** コマンド』

関連情報:

digest コマンド

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・コード・ページ変換テーブル

仮想プリンターの定義と属性

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になります。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になります。**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

項目	説明
-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

項目	説明
<code>/usr/sbin/piocnvt</code>	<code>piocnvt</code> 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

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

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、コロンの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 **pioibe** 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、コロンの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、コロンのフォーマットでの入力ファイルの名前です。コロンのファイルには、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の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

項目	説明
-d <i>QueueDeviceName</i>	仮想プリンター (キュー・デバイス) の名前を指定します。
-n <i>DeviceName</i>	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。 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。
-p <i>DirectoryPath</i>	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。 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。
-q <i>PrintQueueName</i>	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せん。 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。
-s <i>DataStreamType</i>	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。 プリンターのデータ・ストリーム・タイプを指定します。例のデータ・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。
-t <i>PrinterType</i>	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。 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

項目	説明
/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。

関連資料:

372 ページの『**piobe** コマンド』

関連情報:

mkvirprt コマンド

印刷の管理

プリンター・コロン・ファイルの規則

libqb 内のバックエンド・ルーチン

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

項目	説明
-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

- 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
```

- 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

項目	説明
<code>/usr/lib/lpd/pio/etc/piodmgr</code>	piodmgr コマンドが入っています。
<code>/var/spool/lpd/pio/@local/smit/*</code>	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。

関連資料:

372 ページの『`piobe` コマンド』

関連情報:

印刷の管理

印刷スプーラー

プリンター・バックエンドのプログラム

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

piofontin コマンド

目的

マルチリンガル・フォントのディスクセ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・フォントのディスクセットから、**/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

項目	説明
-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	ディスクセット・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クセット・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

ディスクセット・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クセ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

項目	説明
/usr/sbin/piofontin	piofontin コマンドが入っています。

関連情報:

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

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 **piogetopt** サブルーチン、**piogetstr** サブルーチン、**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

項目

-@ DataBaseFile

説明

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・デバイス名を組み合わせると、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** ディレクトリー内でデータベース・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。

項目	説明
%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

項目	説明
/usr/lpd/pio/etc/pioformat	フォーマッター・ドライバーが入っています。
/usr/lpd/pio/fmtrs/*	フォーマッターが入っています。
/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。

関連資料:

372 ページの『**piobe** コマンド』

374 ページの『**pioburst** コマンド』

関連情報:

仮想プリンターの定義と属性

Printer Addition Management Subsystem: Programming Overview

プリンター・コード・ページ変換テーブル

piofquote 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

項目	説明
標準入力	処理される入力データ・ストリーム。
標準出力	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。

関連資料:

- 372 ページの『**piobe** コマンド』
- 374 ページの『**pioburst** コマンド』
- 377 ページの『**piodigest** コマンド』
- 381 ページの『**pioformat** コマンド』
- 393 ページの『**pioout** コマンド』

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** ファイルに使われる順序と同じです。

フラグ

項目	説明
-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:rkmlxx  4019 (IBM ASCII)
4019lxxa:rkmlxxl  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。

```
piolsvp -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    rkmlxx     4019 (IBM ASCII)
4019lxxa    rkmlxx     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。

```
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      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。

```
piolsvp -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。

```
piolsvp -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

項目

`/usr/lib/lpd/pio/etc/piolsvp`

`/etc/qconfig`

`/var/spool/lpd/pio/@local/custom/*`

`/usr/lib/lpd/pio/etc/*.attach`

説明

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。

関連資料:

372 ページの『**piobe** コマンド』

605 ページの『**qprt** コマンド』

関連情報:

プリンター接続ファイル

プリンター・バックエンドのプログラム

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

項目	説明
-a <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。 <code>key_word = value</code>
-A	-D 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。疑似デバイスを追加します。
-C	疑似デバイスを変更します。
-D	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。
-p <i>PseudoDevice</i>	プリンター接続用の疑似デバイス名を指定します。
-R	疑似デバイスを除去します。

ファイル

項目	説明
<code>/usr/lib/lpd/pio/etc/piomgpdev</code>	piomgpdev 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。

関連資料:

372 ページの『`piobe` コマンド』

605 ページの『`qprt` コマンド』

関連情報:

印刷の管理

印刷スプーラー

プリンター・バックエンドのプログラム

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

項目	説明
-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。 <i>AttachmentType</i> 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。 ファイル データを格納するファイルを指定します。
-c CmdExec	SMIT コマンド cmd_to_execute 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 piomkppq コマンドが使用されます。
-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。
-e	既存の印刷キューをプリンターの出力に使用するように指定します。 -e を指定すると、 piomkapqd コマンドを指定しても新しいキューを作成しません。
-f FileName	出力を格納するファイル名を示します。
-h Header	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。
-i DiscCmd	SMIT コマンド cmd_to_discover 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、 piomkapqd 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。
-o ObjectID	<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。
-p Printer	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。
-r ParentAdapter	プリンターの親アダプターを指定します。
-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。 <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">• parallel• rs232• rs422
-T TTYName	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。
-v Device	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。 -v フラグを指定すると、 /usr/lib/lpd/pio/predef 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 hp1j-4 [-d lp0] -c /usr/sbin/mkjetd -i /usr/bin/ljsd -o JetDirect -h 'Add New Attachment Type'
```

ファイル

項目

`/usr/lib/lpd/pio/etc/piomkapqd`

説明

`piomkapqd` コマンドが入っています。

関連資料:

372 ページの『`pio` コマンド』

『`piomkpq` コマンド』

関連情報:

印刷の管理

印刷スプーラー

プリンター・バックエンドのプログラム

piomkpq 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

```
piomkpq -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

```
piomkpq -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

```
piomkpq -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

```
piomkpq -A AttachmentType -p PrinterType -d DeviceName -D DataStream -q QueueName
```

説明

piomkpw 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、**piomkpw** 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。**piomkpw** 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

項目	説明
-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 <i>Attribute=Value</i> 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。 <i>AttachmentType</i> 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。 ファイル 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。 <i>Subclass</i> 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。

```
piomkpw -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。

```
piomkpw -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。

```
piomkpw -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。

```
piomkpw -A file -p ibm4019 -d clues -D asc -q baker
```

ファイル

項目	説明
<code>/usr/lib/lpd/pio/etc/piomkpw</code>	piomkpw 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。

関連資料:

372 ページの『`pio` コマンド』

387 ページの『`piomkapqd` コマンド』

関連情報:

印刷の管理

印刷スプーラー

プリンター・バックエンドのプログラム

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 coron ファイルで使用して、**pio** 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

項目	説明
-a <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、 %s または %n\$s printf 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。 -a 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。
-c <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。 -c フラグは、 -n フラグとともに指定しなければなりません。
-n <i>MsgNumber</i>	メッセージ番号を指定します。 -n フラグは、 -c フラグとともに指定しなければなりません。
-s <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。 -s フラグは、 -c と -n フラグの両方とともに指定しなければなりません。
-u <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、コマンドで区切ります。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。 -u 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

項目
/usr/lib/lpd/pio/etc/piomsg

説明
piomsg コマンドが入っています。

関連資料:

372 ページの『piobe コマンド』

関連情報:

printf コマンド

印刷の管理

印刷スプーラー

プリンター・バックエンドのプログラム

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。

項目	説明
PIOTITLE	印刷ジョブの表題。
PIODEVNAME	デバイス名。
PIOQNAME	印刷キューの名前。
PIOQDNAME	キュー・デバイス名。
PIOFROM	印刷ジョブを実行依頼したユーザー。
PIOMAILONLY	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。
PIOTERM	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

項目	説明
-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	<i>Mask</i> として、1 つまたは複数のデバイス・ドライバのエラー・フラグ名をコンマ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ることである場合、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。 <i>InterventionRequiredUser</i> パラメーターには、1 つまたは複数のユーザー名をコンマ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、 <i>TextString</i> 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。

関連資料:

372 ページの『piobe コマンド』

374 ページの『pioburst コマンド』

377 ページの『piodigest コマンド』

関連情報:

Printer Addition Management Subsystem: Programming Overview

プリンター・コード・ページ変換テーブル

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。それは **mkvirprt** コマンドの逆と考えることができ、**chvirprt** コマンドで表示して、次に **piopredef** コマンドで指定して、サポートされていないプリンターに事前定義された定義を作成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

項目	説明
-d <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ーパーラーを <i>QueueDeviceName</i> 変数で指定します。
-q <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ーパーラーを <i>PrintQueueName</i> 変数で指定します。
-r	-s フラグと -t 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。
-s <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。 asc IBM 拡張 ASCII gl Hewlett-Packard GL pcl Hewlett-Packard PCL ps PostScript 630 Diablo 630 855 Texas Instruments 855
-t <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

項目	説明
<code>/etc/piopredef</code>	piopredef コマンドが入っています。
<code>/usr/lpd/pio/predef/*</code>	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。
<code>/var/spool/lpd/pio/@local/custom/*</code>	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:

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・コロン・ファイルを使用したプリンターの追加

Printer Addition Management Subsystem: Programming Overview

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・ライン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いけない。
2. 現在の **user:group** が、**pkgmap** ファイル内で指定された **user:group** と一致していなければならない。
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いけない。

-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

項目	説明
-d Device	<i>Device</i> 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。 <i>Device</i> 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 <i>Response</i> を識別します。 <i>Pkginst</i> がパッケージである場合、 <i>Response</i> 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、 <i>Pkginst</i> が SIP である場合、 <i>Response</i> 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 <i>Admin</i> 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。 <i>Pkginst</i> 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。
<i>Pkginst</i>	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、 <i>Pkginst</i> 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。
	<i>Pkginst</i> 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・ラインを囲んで ' <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。
-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 <i>Spool</i> に読み込みます。

特別な注意

-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・ラインに *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・ライン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。

項目	説明
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

項目
/var/sadm/install/admin/default
/var/sadm/install/logs/pkginst.log
/var/spool/pkg

説明
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
デフォルトのスプール・ディレクトリー

関連資料:

『pkgask コマンド』
401 ページの『pkgchk コマンド』
405 ページの『pkgmk コマンド』
407 ページの『pkgparam コマンド』
411 ページの『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。

項目	説明
/ptfvars	パッケージに関する変数が含まれます。
/fileinfo	パッケージに関するチェックサム情報が含まれます。
/oldfiles	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

項目	説明
-d <i>Device</i>	<i>Device</i> でパッケージの要求スクリプトを実行します。 <i>Device</i> 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 <i>Response</i>	ファイルまたはディレクトリー <i>Response</i> 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)。 <i>Pkginst</i> がパッケージである場合、 <i>Response</i> 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、 <i>Pkginst</i> が SIP である場合、 <i>Response</i> にはディレクトリーを指定しなければなりません。
<i>Pkginst</i>	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、 <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・ラインを囲んで **'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・ラインに *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・ラインで指定されていなければ参照できません。

終了状況

このコマンドは、以下の終了値を戻します。

項目	説明
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

項目
/var/spool/pkg

説明
デフォルトのスプール・ディレクトリー

関連資料:

- 396 ページの『`pkgadd` コマンド』
- 『`pkgchk` コマンド』
- 407 ページの『`pkgparam` コマンド』
- 413 ページの『`pkgtrans` コマンド』
- 411 ページの『`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・ライン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

項目	説明
-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探索し、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 <i>Pkgmap</i> 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 フラグを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 フラグを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 <i>Pkgmap</i> 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 <i>pkgmap</i> ファイル <i>Pkgmap</i> と対比するように要求します。
-e	指定された <i>pkgmap</i> 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、 <i>Envfile</i> として指定された <i>pkginfo</i> ファイルを使用するように要求します。
<i>Pkginst</i>	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、 <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・ラインを開んで ' <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。
-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。

項目	説明
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

項目	説明
<code>/usr/sbin/pkgchk</code>	<code>pkgchk</code> コマンドが入っています。

関連資料:

- 396 ページの『`pkgadd` コマンド』
- 399 ページの『`pkgask` コマンド』
- 『`pkginfo` コマンド』
- 405 ページの『`pkgmk` コマンド』
- 411 ページの『`pkgrm` コマンド』

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

項目	説明
-q	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 -x 、 -l 、 -p 、および -i 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。)
-x	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。
-l	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。
-r	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。
-p	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。
-i	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。
-a Arch	パッケージのアーキテクチャーを Arch と指定します。
-P Path	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。
-vVersion	パッケージのバージョンを Version 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。
-c Category . . .	<p>カテゴリー Category 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 pkginfo 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、pkginfo 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。</p> <p>指定されたカテゴリーが "set" である場合、pkginfo 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。</p> <p><i>Pkginst</i> 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。)</p> <p>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・ラインを囲んで '<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。</p> <p><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。</p>
-d Device	<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 (/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。

終了状況

このコマンドは、以下の終了値を戻します。

項目	説明
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

項目	説明
<code>/var/spool/pkg</code>	デフォルトのプール・ディレクトリー

関連資料:

396 ページの『`pkgadd` コマンド』

『`pkgmk` コマンド』

411 ページの『`pkgrm` コマンド』

関連情報:

`pkginfo` コマンド

`setinfo` コマンド

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

項目	説明
<code>-a Arch</code>	<code>Arch</code> を使用して <code>pkginfo</code> 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。
<code>-b BaseDir</code>	指定された <code>BaseDir</code> 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。
<code>-c</code>	情報でないファイルを圧縮します。 <code>-c</code> を使用する場合は <code>-r</code> オプションも指定する必要があります。 <code>RootPath</code> 指定より上の相対パスを参照する <code>Prototype</code> ファイル内のエントリーは、圧縮されません。 <code>pkgmk</code> 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)、 <code>pkgadd</code> コマンドにより圧縮解除されません。
<code>-d Directory</code>	<code>Directory</code> 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。

項目	説明
-f <i>Prototype</i>	ファイル <i>Prototype</i> 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。
-l <i>Limit</i>	pkgproto を使用して <i>Prototype</i> 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。
-o	<i>Limit</i> 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。
-p <i>PStamp</i>	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。
-r <i>RootPath</i>	pkginfo ファイル内の実動スタンプ定義を <i>PStamp</i> を使用してオーバーライドします。
-v <i>Version</i>	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 <i>RootPath</i> に、 <i>Prototype</i> ファイル内のソース・パス名を付加します。
<i>Variable=Value</i>	<i>Version</i> 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。
<i>PkgInst</i>	指定された変数をパッケージ環境に入れます。 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/lsp 555 bin bin
f example /example/pkgmk/sbin/luser 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

項目	説明
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

項目	説明
/usr/sbin/pkgmk	pkgmk コマンドが入っています。

関連資料:

- 396 ページの『**pkgadd** コマンド』
- 399 ページの『**pkgask** コマンド』
- 409 ページの『**pkgproto** コマンド』
- 411 ページの『**pkgrm** コマンド』

関連情報:

installp コマンド

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・ラインで要求された 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・ライン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

項目	説明
-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 <i>Device</i>	<i>Pkginst</i> が保管されている <i>Device</i> を指定します。 <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。
-f	コマンドに、 <i>File</i> 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に使用できます。
<i>Pkginst</i>	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、 <i>Pkginst.*</i> 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・ラインを囲んでください。
<i>Param</i>	値を表示すべき特定のパラメーターを定義します。
-P <i>path</i>	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。

項目	説明
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

項目	説明
<code>/var/spool/pkg</code>	デフォルトのスパール・ディレクトリー
<code>/usr/sbin/pkgparam</code>	<code>pkgparam</code> コマンドが入っています。

関連資料:

413 ページの『`pkgtrans` コマンド』

関連情報:

`pkginfo` コマンド

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・ラインでパス が指定されていないと、標準入力がパス のリストと考えられます。コマンド・ライン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 (*f*type=s) にシンボリック・リンク・エンターリーを作成します。**-i** フラグを使用する場合、**pkgproto** はシンボリック・リンク (*f*type=f) にファイル・エンターリーを作成します。プロトタイプ・ファイルを編集して、*v* (揮発 (*v*olatile))、*e* (編集可能 (*e*ditable))、または *x* (排他的ディレクトリー (*e*xclusive 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、 **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:

1. 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。
2. 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、 **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。
3. 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

項目	説明
-i	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。
-c Class	すべてのパスのクラスを Class にマップします。
Path1	オブジェクトが配置されるディレクトリーのパス。
Path2	Path1 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。

1.

```
$ 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
```

2.

```
$ 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
```

3.

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

項目	説明
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

項目	説明
<code>/usr/sbin/pkgproto</code>	pkgproto コマンドが入っています。

関連資料:

405 ページの『**pkgmk** コマンド』

407 ページの『**pkgparam** コマンド』

413 ページの『**pkgtrans** コマンド』

関連情報:

pkginfo コマンド

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・ライン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

項目	説明
-n	非対話式モードを使用可能にします。対話が必要な場合には、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 <i>Admin</i> を定義します。 (<i>Admin</i> 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、 <i>admin</i> 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 <i>/var/sadm/install/admin</i> 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 <i>Spool</i> 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、 <i>Pkginst</i> のすべてのインスタンス化を指します。) <i>Pkginst</i> 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・ラインを囲んで ' <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。

終了状況

このコマンドは、以下の終了値を戻します。

項目	説明
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

項目	説明
/usr/sbin/pkgrm	pkgrm コマンドが入っています。

関連資料:

396 ページの『**pkgadd** コマンド』

399 ページの『**pkgask** コマンド』

『**pkgtrans** コマンド』

関連情報:

pkginfo コマンド

pkgmap コマンド

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

csh からは **pkgtrans** を実行できません。

フラグ

項目	説明
-i	pkginfo ファイルと <i>Pkgmap</i> ファイルをコピーします。パッケージ・カテゴリーがセット・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。

項目	説明
-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 <i>Device2</i> 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、 <i>Device2</i> 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。
<i>Device1</i>	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、 <i>Device2</i> に収められます。 <i>Device1</i> が通常ファイルまたはディレクトリである場合、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。
<i>Device2</i>	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。 <i>Device2</i> が通常ファイルまたはディレクトリである場合、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。
<i>Pkginst</i>	<i>Device1</i> 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。 <i>Pkginst.*</i> 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、 <i>Pkginst</i> 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。

項目	説明
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

項目	説明
<code>/usr/sbin/pkgtrans</code>	<code>pkgtrans</code> コマンドが入っています。

例

1. 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
```

2. `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
```

関連資料:

396 ページの『`pkgadd` コマンド』

399 ページの『`pkgask` コマンド』

403 ページの『`pkginfo` コマンド』

411 ページの『`pkgrm` コマンド』

関連情報:

`pkginfo` コマンド

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` ディレクトリー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

項目	説明
-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、失われることとなります。
-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 <i>fstype</i>	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。 <i>fstype</i> 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 <i>flag</i>	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しないか、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 <i>seq_no</i>	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。 <i>seq_no</i> は、AIX エラー・ログ・ファイル内に保管されている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。
- 416 AIX バージョン 7.2: コマンド・リファレンス 第 4 巻 (n から r)

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

- 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。`enq` コマンドを参照し、`-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

項目	説明
<code>-fr=X</code>	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 <code>X</code> 個のフレームがプロットされますが、この <code>X</code> 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 <code>fr=5</code> が <code>plotgbe</code> バックエンドに渡されることになります。
<code>-noin</code>	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、 <code>plotgbe</code> 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。 <code>-noin</code> 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

- 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

項目	説明
<code>/usr/lpd/plotgbe</code>	<code>plotgbe</code> コマンドが入っています。

関連資料:

『`plotlbe` コマンド』

592 ページの『`qdaemon` コマンド』

関連情報:

`enq` コマンド

印刷の管理

5080 によるプロッター・サポートの追加

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・ライン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

項目	説明
-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

項目	説明
<code>/usr/lpd/plotlbe</code>	plotlbe コマンドが入っています。

関連資料:

417 ページの『`plotlbe` コマンド』

関連情報:

`enq` コマンド

印刷の管理

pmctl コマンド

目的

パフォーマンス・モニター・イベントの生成を開始、リセット、または停止します。

構文

```
pmctl [ { [ -E [ mode ] ] [ -f interval ] { [ -y command ] } } | [ -h ] | [ -r ] | [ -S ] ] [ -s ] [ { -a -y command [ -f interval ] } ]
```

説明

pmctl コマンドは、PMAPI サブシステムでパフォーマンス・モニター・イベントの生成を開始、リセット、または停止して、**tprof -E** コマンドを使用した人手によるオフライン・モードをサポートします。また、PMAPI サブシステムの現在の状況もレポートします。

フラグ

項目	説明
-a	大規模ページ分析をオンにします。
-E [mode]	イベント・ベースのプロファイル作成を可能にします。以下のいずれかのモードを指定できます。
	PM_event プロファイル作成対象のハードウェア・イベントを指定します。 -E フラグに対してモードが無指定の場合、デフォルト・イベントは、プロセッサ・サイクル (PM_CYC) です。
	EMULATION エミュレーション・プロファイル作成モードを使用可能にします。
	ALIGNMENT アライメント・プロファイル作成モードを使用可能にします。
	ISLBMISS インストラクション・セグメント・ルックアサイド・バッファ・ミスのプロファイル作成モードを使用可能にします。
	DSLBMISS データ・セグメント・ルックアサイド・バッファ・ミスのプロファイル作成モードを使用可能にします。
-f interval	使用すべきサンプリング間隔を指定します。 <ul style="list-style-type: none">プロセッサ・サイクルの場合、EMULATION、ALIGNMENT、ISLBMISS、および DSLBMISS の各イベントでは、1 から 500 ミリ秒を指定します (デフォルトは 10 ミリ秒)。パフォーマンス・モニター・イベントの場合、10000 から最大 MAXINT 発生数を指定します (デフォルトは 10000)。 -f フラグを -y フラグと一緒に使用すると、他のパフォーマンス・モニター・イベントの場合は 1 から最大 MAXINT 発生数まで指定します (デフォルトは 10000)。
-h	man ページ情報を表示します。
-r	PMAPI サブシステムの解放とリセットを行います。
-S	パフォーマンス・モニター・イベントの生成を停止します。
-s	PMAPI サブシステムの現在の状況を表示します。
-y command	イベント・ベースのプロファイル作成をオンにするのは、指定コマンドとその子孫に限定されます。

例

- パフォーマンス・モニター・イベントの生成を停止するには、以下のコマンドを入力します。

```
pmctl -S
```
- パフォーマンス・モニター・イベントの生成をリセットするには、以下のコマンドを入力します。

```
pmctl -r
```

3. PMAPI サブシステム内の現在の状況をレポートするには、以下のコマンドを入力します。

```
pmctl -s
```

4. パフォーマンス・モニター・イベントの生成を開始するには、以下のコマンドを入力します。

```
pmctl -E
```

5. 指定の **workload** コマンドとその子孫に対してのみパフォーマンス・モニター・イベントの生成を開始するには、以下のコマンドを入力します。

```
pmctl -E -y workload
```

6. 人手によるオフライン・モードで **tprof -E**をサポートするには、以下のコマンドを入力します。

```
trace -adf -o mydata.trc
trcon
pmctl -E
sleep 10; trcstop
gensyms > mydata.syms
tprof -suker mydata
```

7. 指定の **workload** コマンドとその子孫に対してプロファイル作成するための人手によるオフライン・モード状態で、**tprof -E**をサポートするには、以下のコマンドを入力します。

```
trace -adf -o mydata.trc
trcon
pmctl -E -y workload
trcstop
gensyms > mydata.syms
tprof -suker mydata
```

関連情報:

tprof コマンド

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.pmap**i 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

項目	説明
-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_cycle コマンド

pmlist 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

```
pmlist [ -h ]
```

```
pmlist -l [ -o t | c | x ]
```

```
pmlist { -s | -e ShortName | -c Counter [ ,event ] | -g Group | -S Set | -D DerivedMetricID | -m MetricGroup | -V Variable } [ -p ProcessorType ] [ -s ] [ -d ] [ -o t | c | x ] [ -f Filter ]
```

説明

pmlist 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。
- 派生メトリック・ファイル内の変数をリストします。

フラグ

項目	説明
-c -1	すべてのカウンターのすべてのイベントを表示します。
-c Counter	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。
-c Counter,Event	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。
-d	イベントの詳細記述を表示します。
-D -1	サポートされる派生メトリックをすべて表示します。
-D DerivedMetricID	指定された <i>DerivedMetricID</i> を表示します。
-e ShortName	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。
-f v,u,c	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、 v (検査済み)、 u (未検査)、および c (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 v,u,c です。
-g -1	すべてのイベント・グループのリストを表示します。
-g Group	指定したイベント・グループ <i>Group</i> を表示します。
-h	pmlist コマンドのヘルプ情報を表示します。
-l	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。
-m -1	メトリック・グループ別にすべての派生メトリックをリストします。
-m MetricGroup	指定された <i>MetricGroup</i> に関連した派生メトリックをすべて表示します。
-o t c x	pmlist コマンドの出力形式を指定します。有効な出力形式は、 t (テキスト形式)、 c (CSV 形式)、 x (XML 形式) のいずれかとして指定します。デフォルトの出力形式はテキストです。
-p ProcessorType	プロセッサ・タイプを指定します。
-s	プロセッサ情報の要約を表示します。
-S -1	サポートされるイベント・セットをすべて表示します。
-S Set	指定されたイベント <i>Set</i> を表示します。
-V -1	派生メトリックの計算に使用する変数をすべて表示します。
-V Variable	指定された変数を表示します。

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
```

6. 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:

POWERCOMPAT イベント・リスト

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

項目	説明
-dst	削除されるべき pmtu エントリーの宛先を指定します。
-gw	削除されるべき pmtu エントリーのゲートウェイを指定します。
-inet6	IPv6 pmtu 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

項目	説明
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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**Issecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

1. 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

項目	説明
<code>/usr/sbin/pmtu</code>	<code>pmtu</code> コマンドが入っています。

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、標準入力で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

項目	説明
-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

項目	説明
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。以下のリストは、**imap** サービス用の */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

項目	説明
<i>/usr/sbin/pop3d</i>	pop3d コマンドが入っています。
html	

関連情報:

imapd コマンド

pop3ds デーモン

目的

TLS/SSL により 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

pop3ds [-c]

説明

pop3ds 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。**pop3ds** 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。

pop3ds コマンドは、**sendmail** および **bellmail** で構成された、既存のメール・インフラストラクチャーと連動します。

フラグ

項目	説明
-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

項目	説明
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

pop3ds 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。**pop3ds** 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。以下のリストは、**imap** サービス用の */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: **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

項目	説明
<code>/usr/sbin/pop3ds</code> html	<code>pop3ds</code> コマンドが入っています。

関連情報:

imapds コマンド

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

項目	説明
<code>inetd.conf</code> <code>/etc/rpc</code>	RPC デーモンおよびその他の TCP/IP デーモンを始動します。 サーバー名およびそれに対応する <code>rpc</code> プログラム番号と別名のリストが入っています。

関連資料:

892 ページの『`rpcinfo` コマンド』

関連情報:

`inetd` コマンド

ネットワーク・ファイルシステム (NFS) の概要

TCP/IP プロトコル

NFS コマンド

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
```

フラグ

項目	説明
-c <i>monitor</i>	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 <i>mir_modem</i>	ダイヤ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 <i>monitor</i>	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 <i>target</i>	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。

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
<code>/usr/share/modems/mir_modem</code>	ダイヤルインの設定のモデム構成ファイル例。
<code>/usr/sbin/portmir</code>	コマンド・ファイルが入っています。

関連情報:

chdev コマンド

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

項目	説明
-alias <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。 post コマンドは、自動的に /etc/mh/MailAliases ファイルを検索します。
-filter <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。
-format	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。
-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。
-msgid	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。
-nofilter	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。
-noformat	受信側のアドレスのフォーマットを変更しません。
-nomsgid	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。
-noverbose	メッセージを sendmail 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。
-nowatch	sendmail 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。
-verbose	メッセージを sendmail 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。
-watch	メッセージを sendmail 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。
-width <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

項目	説明
/etc/mh/MailAliases	デフォルトのメール別名が入っています。
/etc/mh/mtstailor	MH コマンド定義が入っています。

関連情報:

sendmail コマンド

spost コマンド

mhmail コマンド

whom コマンド

MH の **mtstailor** ファイル

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。最大受信再構造単位 (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

項目	説明
authenticate pap chap client server demand	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。
ip ipv6 ip ipv6 connect "ConnectorProgram"	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。
inactive Seconds	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。
multilink	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。
nodaemon	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。
peer pap chap remote HostName	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 に対する入力は、 /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。

項目	説明
<code>user Name</code>	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、 <code>UserName RemoteHostName Password</code> に対する入力は、 <code>/etc/ppp/pap-secrets</code> 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。

項目	説明
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: 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なりたい場合は、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

項目	説明
<code>/usr/sbin/pppattachd</code>	<code>pppattachd</code> デーモンが入っています。
<code>/etc/ppp/attXXX.pid</code>	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 <code>uucp</code> グループに属さなければなりません。

関連資料:

『pppcontrold デーモン』

440 ページの『pppdial コマンド』

関連情報:

syslog コマンド

profile コマンド

非同期 PPP サブシステム

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 <i>name</i>	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。
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。

項目	説明
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。
-negacf	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。
-negprotocolcompress	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

interface	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。
ip および ipv6	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、 local_ip 、 local_ip6 、 remote_ip 、および remote_ip6 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。
サーバー	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られます。**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られます。結局、4 つすべてが入れられるのは、「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 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 ::0007:0000:0000:4446
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。

項目	説明
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。

例

/ect/pppp/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 0xffffffff00
pp4: flags=31<UP,POINTOPOINT,NOTRAILERS>
inet 129.35.130.45 --> 129.35.131.212 netmask 0xffffffff00
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

項目	説明
<code>/usr/sbin/pppcontrold</code>	pppcontrold 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 (lcp_config 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 (if_conf 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>	pppcontrold プロセス id が入っています。
<code>/etc/ppp/ppp.conf</code>	strload コマンドへの入力が入っています。

関連資料:

432 ページの『`pppattachd` デーモン』

『`pppdial` コマンド』

関連情報:

`startsrc` コマンド

非同期 PPP サブシステム

システム・リソース・コントローラー

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** デー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

項目	説明
<code>-d VerboseFile</code>	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、 pppdial 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、 pppdial コマンドは既存ファイルに出力を追加します。
<code>-f ChatFile</code>	<code>tty</code> 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。
<code>-tTimeOut</code>	チャット・アクティビティーの Expect 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。
<code>-v</code>	<code>syslog</code> 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。

項目	説明
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: 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・ラインに以下のように 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 sword  
pppuserpwd Send the string pppuserpwd  
           pppuserpwd is the password of the user pppuser on the  
           リモート・システム
```

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

項目	説明
<code>/usr/sbin/pppdial</code>	<code>pppdial</code> コマンドが入っています。

関連資料:

432 ページの『`pppattachd` デーモン』

435 ページの『`pppcontrold` デーモン』

関連情報:

`syslog` コマンド

非同期 PPP サブシステム

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

項目	説明
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

項目	説明
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

項目	説明
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。

項目	説明
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

項目	説明
/usr/sbin/pppstat	pppstat コマンドが入っています。

関連資料:

- 440 ページの『**pppdial** コマンド』
- 435 ページの『**pppcontrold** デーモン』
- 432 ページの『**pppattachd** デーモン』

関連情報:

profile コマンド
syslog コマンド

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 | -I pprof.flow | -i tracefile | -d } [ -T bytes ] [ -v ] [ -s ] [ -n ] [ -f ] [ -p ] [ -w ] [ -r
PURR ] [ -@ [ WparList | ALL ]
```

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

項目	説明
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。
pprof.start	引数のない -@ フラグを選択した場合は、WPAR 名も提供されます。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。
pprof.namecpu	引数のない -@ フラグを選択した場合は、WPAR 名も提供されます。各タイプのカーネル・スレッド (すべて同じ名前で行可能) 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 %。
pprof.famind	引数のない -@ フラグを選択した場合は、WPAR 名も提供されます。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。
pprof.famcpu	引数のない -@ フラグを選択した場合は、WPAR 名も提供されます。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。
	引数のない -@ フラグを選択した場合は、WPAR 名も提供されます。

フラグ

項目	説明
-d	ユーザーが、コマンド・ラインから trcon と trcstop を実行するのを待ちます。
-f	pprof.famcpu レポートと pprof.famind レポートだけを生成するように指定します。
-i tracefile	tracefile 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。
-I pprof.flow	以前に生成された pprof.flow からレポートを生成するように示します。 pprof.namecpu レポートだけを生成するように指定します。
-n	pprof.namecpu レポートだけを生成するように指定します。
-p	pprof.cpu レポートだけを生成するように指定します。
-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。
-s	pprof.start レポートだけを生成するように指定します。
-T	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。
-v	冗長モード (エクストラ詳細の表示) を設定します。
-w	pprof.flow だけを生成するように指定します。
-@ [WparList ALL]	WPAR 情報を表示します。
	ALL すべての WPAR をリストします。
	WparList 関心のある WPAR のコンマ区切りリストを指定します。
<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
trcon コマンド
trcstop コマンド

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

項目	説明
-Column	欄数を、 <i>Column</i> 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 フラグが指定されます。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	<i>Page</i> 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、 <i>File</i> 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。

項目	説明
<code>-i[Character][Gap]</code>	<code>Gap+1</code> 、 <code>2*Gap+1</code> 、および <code>3*Gap+1</code> 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。 <code>Gap</code> のデフォルト値は 8 です。 <code>Character</code> 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。
<code>-l Lines</code>	66 行のデフォルトを取り消し、ページの長さを <code>Lines</code> 変数で指定した行数に再設定します。 <code>Lines</code> 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 (<code>-t</code> フラグが有効であるときのように)。
<code>-m</code>	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、 <code>pr</code> コマンドが <code>File</code> 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 <code>-Column</code> フラグとともに使用しないでください。
<code>-n[Character][Width]</code>	<code>Width</code> 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 <code>-m</code> フラグが設定されている場合には出力の各行の最初の <code>Width+1</code> 桁位置に示されます。 <code>Character</code> 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。
<code>-o Offset</code>	<code>Offset</code> 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。 <code>Offset</code> のデフォルト値は 0 です。
<code>-p</code>	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。 <code>pr</code> 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 <code>Enter</code> キーを押すのを待ちます。
<code>-r</code>	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。
<code>-s[Character]</code>	適切な数のスペースで区切る代わりに、 <code>Character</code> 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。 <code>Character</code> 変数のデフォルト値は、ASCII の TAB 文字です。
<code>-t</code>	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。
<code>-w Width</code>	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。 <code>-w</code> オプションも <code>-s</code> 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。 <code>-w</code> オプションを指定せずに <code>-s</code> 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。
<code>-x[Character][Width]</code>	<code>-n</code> 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。

項目	説明
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。

- 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。

- 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。

- 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

項目	説明
<code>/usr/bin/pr</code>	pr コマンドが入っています。
<code>/dev/tty*</code>	メッセージを延期します。

関連資料:

605 ページの『**qprt** コマンド』

関連情報:

cat コマンド

ファイル・コマンド

shells コマンド

ファイルとディレクトリーのアクセス・モード

praliases コマンド

目的

システムのメール別名を表示します。

構文

```
praliases [-C file] [-f file] [key]
```

説明

praliases コマンドは、特定の順序なしに、行ごとに、システムの現行別名を表示します。特別な内部 @: @別名が表示されます (ある場合)。

フラグ

項目	説明
-C file	デフォルトの <code>sendmail</code> 構成ファイルではなく、指定された <code>sendmail</code> 構成ファイルを読み取ります。
-file	<code>sendmail</code> ファイルの構成済み別名ファイルではなく、指定されたファイルを読み取ります。
key	コマンド・ラインで 1 つ以上のキーが指定されている場合、そのキーに一致するエントリを表示します。

注: `praliases` コマンドは、成功時には **0** で、エラーが発生する場合は **>0** で終了します。

ファイル

項目	説明
<code>/etc/mail/sendmail.cf</code>	デフォルトの <code>sendmail</code> 構成ファイルが入っています。

関連情報:

`mailq` コマンド

`sendmail` コマンド

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

項目	説明
<code>/usr/sbin/acct</code>	アカウントティング・コマンドへのパス。
<code>/var/adm/acct/nite</code>	アカウントティング・データ・ファイルが入っています。

関連情報:

システム・アカウントティング
アカウントティング・サブシステムの設定

prdaily コマンド

目的

前日のアカウントティング情報の 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、前日のアカウントティング・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

項目	説明
<code>-c</code>	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、当日のアカウントティング・データだけにしか使用できないことがあります。
<code>-l [mmdd]</code>	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、 <i>mmdd</i> 変数を使用します。
<code>-X</code>	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 <code>/var/adm/acct/sum</code> ディレクトリーではなく <code>/var/adm/acct/sumx</code> ディレクトリーを使用します。

セキュリティー

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

項目	説明
<code>/usr/sbin/acct</code>	アカウントティング・コマンドへのパス。
<code>/usr/sbin/acct/ptelus.awk</code>	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。
<code>/usr/sbin/acct/ptecms.awk</code>	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。
<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>	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

システム・アカウントिंग

アカウントिंग・サブシステムの設定

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。

このコマンドを実行するのに、System Management Interface Tool (SMIT) が使えます。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。

通常環境では、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
```

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. Resource Monitoring and Control (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・ライン・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・ラインに入力したフラグが誤りです。
- 4 コマンド・ラインに入力したパラメーターが誤りです。
- 5 誤ったコマンド・ライン入力により、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 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

/opt/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。

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

項目	説明
+Folder	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。
-header	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。
-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。
-noheader	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。
-noshowproc	/usr/bin/cat コマンドを使用して、前のコマンドをリストします。
-showproc <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。

項目	説明
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するプログラムを指定します。

セキュリティー

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティー」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
<code>\$HOME/.mh_profile</code>	MH ユーザー・プロファイルが入っています。
<code>/usr/bin/prev</code>	<code>prev</code> コマンドが入っています。

関連資料:

63 ページの『next コマンド』

関連情報:

show コマンド
 .mh_alias コマンド
 .mh_profile コマンド
 メール・アプリケーション

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 を戻します。それ以外の場合には、0 (ゼロ) の状況に戻します。

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** 環境変数の値を戻します。以下に例を示します。

```
YOU HAVE NEW MAIL
```

関連情報:

env コマンド
 environment コマンド
 プロファイル概要
 shells コマンド

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。

項目	説明
LANG	LC_ALL および対応する環境変数 (LC_ 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。
LC_ALL	LANG やそのほかの LC_ 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。
LC_CTYPE	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。
LC_MESSAGES	メッセージを書き出す際の言語を決めます。
LC_NUMERIC	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、 e 、 E 、 f 、 g 、および G 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。

項目	説明
¥¥	円記号
¥a	アラート
¥b	バックスペース
¥f	用紙送り
¥n	改行
¥r	復帰
¥t	タブ
¥v	垂直タブ
¥ddd	<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。

項目	説明
-	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 変換の最大有効桁数。
 - 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。

項目	説明
%	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。

項目	説明
f	浮動小数点値または倍精度値を受け取り、それを [-] <i>ddd.ddd</i> . 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 [-] <i>d.dde[+ -]dd</i> 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	値を円記号エスケープ・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。

以下の円記号エスケープ・シーケンスがサポートされます。

- 前述の *Format* パラメーター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** (円記号 **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ー内の残りの文字、残りの文字列パラメーター、および *Format* パラメーターの付加文字を、**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。

項目	説明
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* パラメーターが 3 回使用されています。最後の **%d** 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$s, %4$d:%5$.2d", weekday, month, day, hour, min);  
Sunday, 3. July, 10:02  
(weekday, day. month, hour:min)
```

ファイル

項目	説明
<code>/usr/bin/printf</code>	<code>printf</code> コマンドが入っています。

関連情報:

`/usr/bin/echo` コマンド

`printf` コマンド

入出力処理のプログラマーのための概要

ナショナル・ランゲージ・サポートの概要

probevctl コマンド

目的

ProbeVue 動的トレース・パラメーターおよび ProbeVue セッションの変更と表示を行います。

構文

```
probevctl [ -c attribute = value ] [ -C ] [ -f { milli|micro } ] [ -d sessionID ] [ -l ] [ -n attribute =  
value ] [ -p ] [ -s { probevue_session_id } ] [ -t ] [ -u user-list ] [-T { show|start|stop|reset }]
```

説明

`probevctl` コマンドは、ProbeVue 動的トレース・パラメーター、プロセッサ別のトレース・バッファ・サイズ、固定メモリー消費量、セッションを所有するユーザー、セッションを開始したプロセスの ID、および該当セッションに ProbeVue セッション用のカーネル・プローブがあるかどうかの情報の変更と表示を行います。

構成可能な ProbeVue パラメーターは次のとおりです。

- ProbeVue 状況 (使用可能/使用不可)。
- すべての ProbeVue セッションに割り当てられる最大固定メモリー (MB)。
- 非特権ユーザーの ProbeVue セッションに割り当てられる最大固定メモリー (KB)。トレース・バッファ用メモリーを含む。
- 通常のユーザーに許可される並行 ProbeVue セッションの数。
- プロセッサ別のトレース・バッファのデフォルト・サイズ (KB)。
- トレース・バッファからの読み取りのために通常のユーザーがトレース・コンシューマーに要求できる最小期間 (ミリ秒)。
- ProbeVue バッファがトレース・コンシューマーによって読み取られるデフォルト期間 (ミリ秒)。
- ProbeVue セッションで使用されるプロセッサ別の計算スタックのサイズ (KB)。
- 間隔プローブでグローバル root ユーザーに許可されている最小の時間間隔。
- 動的データ構造に対して割り当てられるメモリーのパーセンテージ。

- プロセッサ別ローカル・テーブルのサイズ (KB)。
 - ページ・フォールトを処理するためのページ・フォールト・コンテキストの数。
 - スレッド・ローカル変数がある場合に、ProbeVue セッションがサポートすべきスレッドの最大数。
 - **net** プローブ・アクションによって使用される、CPU 別のバッファの最大サイズ (バイト単位)。
 - アクションが割り込みコンテキスト内で開始された場合に、**systrace** プローブ・アクションにかかる最大時間 (ミリ秒単位)。
 - アクションが割り込みコンテキスト内で開始された場合に、**sysproc** プローブ・アクションにかかる最大時間 (ミリ秒単位)。
 - アクションが割り込みコンテキスト内で開始された場合に、**io** プローブ・アクションにかかる最大時間 (ミリ秒単位)。
 - **net** プローブ・アクションが割り込みコンテキストにおいて開始されたときに、そのアクションに費やすことができる最大時間 (ミリ秒)。
- l • **CPU-bound interval** プローブ・アクションが割り込みコンテキストにおいて開始されたときに、その
- l アクションに費やすことができる最大時間 (ミリ秒)。

root ユーザーまたは **aix.ras.probevue.manage** 権限を持っているユーザーのみが、ProbeVue パラメータを更新し、すべての ProbeVue セッションを表示できます。それ以外のユーザーは、自分が所有するセッションのみを表示できます。各セッションは次のフォーマットで表示されます。

Sid	Pid	Uid	バッファ・サイズ (バイト)	消費されたメモリー (バイト)	カーネル・プローブ	プロファイル作成
<i>sid</i>	<i>pid</i>	<i>uid</i>	<i>bufsize</i>	<i>memory</i>	yes または no	yes または no

デフォルトでは、ProbeVue は使用可能です。ProbeVue セッションがアクティブなときに ProbeVue を使用不可にしようとするとう失敗します。

フラグ

- c 非ユーザー ProbeVue パラメータを指定します。このフラグの引数は、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。このフラグと一緒に **-p** フラグまたは **-t** フラグのいずれも指定しなかった場合、現在のブート・セッションと次回のブート・セッションの両方で新しい値が有効になります。有効な属性/値 のペアは次のとおりです。

trace= {on | off}

ProbeVue を使用可能にするか使用不可にするかを指定します。

default_buffer_size=default_buffer_size

プロセッサ別のトレース・バッファのデフォルト・サイズ (KB) を指定します。この値は次の 4KB ページに繰り上げて丸めます。

max_total_mem_size=max_total_mem_size

ProbeVue フレームワーク全体で消費できる最大固定メモリー (MB) を指定します。

default_read_rate=default_read_rate

ProbeVue バッファがトレース・コンシューマーによって読み取られるデフォルト期間 (ミリ秒) を指定します。

stack_size=stack_size_in_4Kpages

プロセッサ別の計算スタックのサイズ (KB) を指定します。この値は次の 4KB ページに繰り上げて丸めます。

local_table_size=number

プロセッサ別ローカル・テーブルのサイズ (KB) を指定します。ローカル・テーブル用に割り当てられたスペースの半分は、一時的な文字列で使用されます。デフォルト値は 4 KB に設定されています。

min_interval=interval in ms

間隔プローブでグローバル root ユーザーに許可されている最小の時間間隔を指定します。

num_pagefaults=number

ページ・フォールトを処理するためのページ・フォールト・コンテキストの数を指定します。指定されたページ・フォールト・コンテキストの数は、ProbeVue フレームワークの初期化中に事前割り振りされます。

num_threads_traced=number

スレッド・ローカル変数がある場合に、ProbeVue セッションがサポートできるスレッドの最大数を指定します。ProbeVue フレームワークは、この属性によって指定されているスレッドの最大数について、セッションの開始時にすべてのスレッド・ローカル変数を事前割り振りします。

max_net_buf_size=number

net プローブ・アクションによって使用される、CPU 別のバッファの最大サイズ (バイト単位) を指定します。

max_intr_systcrpb_time=number

systrace プローブ・アクションが割り込みコンテキスト内で開始された場合に、プローブ・アクションの実行にかかる時間の最大量 (ミリ秒単位) を指定します。

max_intr_sysprocrb_time=number

sysproc プローブ・アクションが割り込みコンテキスト内で開始された場合に、プローブ・アクションの実行にかかる時間の最大量 (ミリ秒単位) を指定します。

max_intr_ioprpb_time=number

io プローブ・アクションが割り込みコンテキスト内で開始された場合に、プローブ・アクションの実行にかかる時間の最大量 (ミリ秒単位) を指定します。

max_intr_netprpb_time=number

net プローブ・アクションが割り込みコンテキスト内で開始された場合に、プローブ・アクションの実行にかかる時間の最大量 (ミリ秒単位) を指定します。

async_stats_fetch_interval=number

システム統計をフェッチするために、非同期フェッチ間隔をミリ秒単位で指定します。この属性は、すべての ProbeVue セッションに適用されるグローバル値です。ProbeVue は、この値に基づいて非同期フェッチ間隔を設定します。チューナブルのデフォルト値は 1000 ミリ秒です。チューナブル値の変更は、実行中のセッションには影響せず、指定された値は、新規セッションに対してのみ使用されます。

fetch_stats_async_only={yes | no}

同期フェッチが可能な場合でも、すべてのシステム統計を非同期モードでフェッチする必要があることを指定します。このパラメーターのデフォルト値は no です。チューナブル値の変更は、実行中のセッションには影響せず、指定された値は、新規セッションに対してのみ使用されます。

| **max_intr_cpuboundprb_time=number**

| これは、**CPU-bound interval** プローブ・アクションが割り込みコンテキストにおいて開始さ
| れたときに、そのプローブ・アクションの実行に費やすことができる最大時間 (ミリ秒) を指
| 定します。

- C ProbeVue セッション・チューナブルを初期値に設定します。-C オプションは、システムの現行構成を使用して、設定する構成の初期値 (High Config または Low Config) を判別し、現行パラメーターと次のブート・パラメーターの両方を更新します。チューナブルの高低の構成値について詳しくは、**ProbeVue** 動的トレース機能トピックを参照してください。

注: -C フラグを使用する前に、アクティブな ProbeVue セッションがないことを確認してください。

-d *sessionId*

指定されたセッションについて使用可能にするプローブのリストを表示します。セッション ID として all を指定すると、ユーザーが表示できるすべての ProbeVue セッションについて、プローブが表示されます。ProbeVue セッションおよび関連するセッション ID のリストは、**probevctrl** コマンドを使用して入手することができます。

- f プローブ・アクションの時間消費データを表示するフォーマットを指定します。-f オプションを -T オプションと一緒に使用すると、特定のセッションに関するアクションを表示できます。有効なフォーマットは次のとおりです。

milli 時間をミリ秒単位で表示します。

micro 時間をマイクロ秒単位で表示します。

デフォルトのフォーマットは **milli** です。

- l ProbeVue 構成パラメーターの現在の値をリストします。-p フラグまたは -t フラグを指定しなかった場合は、現在のブート・セッションのパラメーター値が表示されます。
- n 通常のユーザーのための構成可能パラメーターを指定します。このオプションの引数は、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。このフラグと一緒に -p フラグまたは -t フラグのいずれも指定しなかった場合、現在のブート・セッションと次のブート・セッションの両方で新しい値が有効になります。有効な属性/値 のペアは次のとおりです。

max_mem_size=max_mem_size

ProbeVue セッションで消費される最大固定メモリー (MB) を指定します。

max_sessions=max_sessn

許可される並行セッションの最大数を指定します。

min_read_rate=min_read_rate

トレース・バッファからの読み取りのために通常のユーザーがトレース・コンシューマーに要求できる最小期間 (ミリ秒) を指定します。

pin_mem_dvar_pc=pin_mem_dvar_pc

動的型変数について動的データ構造に割り当てることができるメモリーのパーセンテージを指定します。このメモリーは、スタック・トレースおよび結合配列型の動的変数に対して使用できます。このパラメーターの値は、10 から 100 の範囲に設定されます。デフォルト値は 50 です。

- p 次のブートのデフォルト値を更新して表示することを指定します。

-s *probevue_session_id*

アクションを実行するセッションを指定します。このフラグは、-T フラグと一緒に使用する必要があります。

-u ProbeVue セッションのリストが必要なユーザーのコンマ区切りリストを指定します。 **-u** フラグを指定しない場合は、ユーザーが表示できるすべての ProbeVue セッションが表示されます。

aix.ras.probevue.manage 権限を持つユーザーは、システム内のすべての ProbeVue セッションを表示できます。この権限のないユーザーは、自分が所有する ProbeVue セッションのみを表示できます。

-t 現在のブート・セッションのデフォルト値を更新して表示することを指定します。

-T show | start | stop | reset

-s フラグで指定されたセッションのプロープ・アクションのプロファイル作成を表示、開始、停止、またはリセットするかどうかを指定します。このフラグと一緒に **-s** フラグを使用する必要があります。

例

1. 次回のブートのデフォルト・バッファ・サイズを変更し、動的トレースをオンにするには、次のように入力します。

```
probevctrl -c trace=on,default_buffer_size=8 -p
```

または

```
probevctrl -c "trace=on default_buffer_size=8" -p
```

2. 次回のブートの ProbeVue 構成をリストするには、次のように入力します。

```
probevctrl -l -p
```

3. 現在の ProbeVue 構成をリストするには、次のように入力します。

```
probevctrl -l -t
```

4. すべての ProbeVue セッションをリストするには、次のように入力します。

```
probevctrl
```

5. ユーザー `guest` が所有するすべての ProbeVue セッションをリストするには、次のように入力します。

```
probevctrl -u guest
```

6. 次回のブートについて動的データ構造 (スタック・トレースおよび結合配列) に割り当てられる `pinned` (滞留) されるメモリのパーセンテージをデフォルトの 50 -75 から増やすには、次のように入力します。

```
probevctrl -n pin_mem_dvar_pc = 75
```

関連資料:

『probevue コマンド』

probevue コマンド

目的

動的トレース・セッションを開始します。このコマンドは、動的トレース・セッションを開始しなくても、ヘッダー・ファイルをプリプロセスして終了することができます。

構文

```
probevue [ -c "{ timestamp = { 0 | 1 | 2 } thread = { on | off } tid = { t1, ... } pid = { p1, ... }  
abs_mem_for_dvars=memory_in_KB num_threads_traced=number_of_thread_to_trace}" ] [-d] [-i  
Async_Fetch_Interval ] [-e Pinned_memory_dvar_percent ] [-f to_print_time_profile_data_milli_or_micro ] [
```

```
-I Include_file1, ... ] [ -K ] [ -o Output_file ] [-q info={none|normal|detail}] [ -s Buffer_size ][ -t Interval ][ -T ] [-u] [ -X Program_name [ -A "Arguments_to_program" ] ] [ Script_name [ Arguments_to_script ] ]
```

```
probevue [ -P C ++_header_file ]
```

```
probevue [ -l "{ syscall | syscallx | syscallx32 | syscallx64 | interval | systrace | sysproc | io | net}" ]
```

説明

probevue コマンドは、オペレーティング・システムおよびユーザー・プログラムを分析するために、ユーザー指定のプローブを動的に使用可能にし、プローブの起動時にプローブに関連付けられたアクションを開始し、収集したトレース・データを表示します。

vue スクリプトで **probevue** コマンドを指定すると、このコマンドはスクリプトに指定されたトレースを使用可能にし、トレース出力を生成します。

-P オプションが C++ ヘッダー・ファイルと一緒に指定されると、このコマンドはプリプロセスされた暗号化 C ヘッダー・ファイルを生成します。暗号化 C ヘッダー・ファイルはさらに、**probevue** コマンドの **-I** オプションを使用することにより、C++ アプリケーションをプローブするのに使用できます。

probevue コマンドおよび **vue** スクリプトへの引数は、コマンド・ラインの代わりにスクリプトで指定できます。このスクリプトを使用することで、同じ引数を使用してスクリプトを繰り返し実行できます。**vue** 言語構成要素 **#VUE_CMD_ARGS** は、ファイルで **probevue** コマンドへの引数を指定するために使用できます。また、言語構成要素 **#VUE_SCRIPT_ARGS** は、**vue** スクリプトへの引数を指定するために使用できます。

注:

- 引数がスクリプトで指定される場合、**vue** 構成体のすべての引数が同じ行になければなりません。
- **probevue** コマンドへの引数がコマンド・ラインとスクリプトで指定されている場合、コマンド・ラインの引数のみが考慮され、スクリプト内のコマンド引数はすべて無視されます。この処理は、**vue** スクリプトの引数にも適用されます。
- **#VUE_CMD_ARGS** 構成体および **#VUE_SCRIPT_ARGS** 構成体は、標準入力をサポートしません。
- **probevue** コマンドの動的メモリー所要量は、CPU の製品数と、CPU 当たりのトレース・バッファ・サイズ (**probevctrl** コマンドの **default_buffer_size** チューナブル・パラメーターの値) に比例します。このため、多数の CPU があり、CPU 当たりのバッファ値が高いシステムでは、**probevue** コマンドが、**ulimit** パラメーターによって設定されたメモリー制限を超える可能性があります。そのようなシナリオでは、その制限を超えるように、**probevue** コマンドで **-u** フラグを設定して実行してください。

フラグ

項目	説明
-A "Arguments_to_program"	-X フラグを使用して指定したプログラムに対する引数を指定します。アプリケーションの引数が複数ある場合は、各引数を引用符で囲みます。
-c	<p>トレース・データをフォーマットする方法を指定します。このオプションの引数はそれぞれ引用符で囲み、引数をスペースで区切る必要があります。オプションは次のとおりです。</p> <p>timestamp={0 1 2}</p> <p>トレース・レポート内の 1 つのイベントに関連付けられたタイム・スタンプの報告を制御します。次の値のいずれかを指定します。</p> <ul style="list-style-type: none"> 0 各メッセージについて、トレース開始時に相対的なタイム・スタンプ (秒およびマイクロ秒) を表示します。トレース出力の最初の行には、個別タイム・スタンプを測る起点となる基本時刻が示されます。 1 各メッセージについて、メッセージの作成にかかる実際の時間を表示します。 2 各メッセージについて、メッセージの作成にかかる実際の時間を <i>printf's %A</i> フォーマットで表示します。 <p>注: 両方のオプションが必要な場合は、0,1 と入力してください。0,1 の途中にスペースは入れません。</p> <p>thread={on off}</p> <p>各メッセージを生成したスレッド ID を、そのメッセージと共に表示します。デフォルト値は off です。</p> <p>pid={p1,..}</p> <p>指定されたプロセスによって生成されたメッセージのみを表示します。 注: スレッドが属するプロセスをトレース・コンシューマーが識別しようとする前にそのスレッドが停止した場合、または指定したプロセスが既に存在しない場合は、メッセージをプロセス ID でフィルタリングしても、コンシューマーはそのプロセス内のスレッドによって生成されたメッセージを表示できません。</p> <p>tid={t1,..}</p> <p>指定されたスレッドによって生成されたメッセージのみを表示します。</p> <p>abs_mem_for_dvars=memory_in_KB</p> <p>動的型変数に割り当てられる固定メモリーをキロバイト (KB) 単位で指定します。このオプションは -e オプションと相互に排他的です。</p> <p>num_threads_traced=number_of_thread_to_trace</p> <p>セッションにスレッド・ローカル変数がある場合に、現行の ProbeVue セッションがサポートできるスレッドの最大数を指定します。この値は、対応するグローバル ProbeVue チューナブル値をオーバーライドします。</p>
-d	セッションについて使用可能にするプローブのリストを表示します。
-e Pinned_memory_dvar_percent	動的型変数に割り振られる動的データ構造メモリーのパーセンテージを指定します。パーセンテージとして最小 10、最大 100 の値を指定できます。
-f	<p>プローブ・アクションにかかった時間を表示するフォーマットを指定します。サポートされるフォーマットは、次のとおりです。</p> <p>milli 時間をミリ秒単位で表示します。</p> <p>micro 時間をマイクロ秒単位で表示します。 デフォルトのフォーマットは milli です。</p>
-i Async_Fetch_Interval	probevue コマンドについて非同期統計がフェッチされる回数を指定します。このオプションは、 probevue コマンドのグローバル間隔時間をオーバーライドします。最小間隔は 100 ミリ秒です。
-I Include_file1	後処理済みヘッダー・ファイル (つまり C プリプロセッサ演算子のないヘッダー・ファイル) として指定されたファイルを使用します。これは vue スクリプトのコンパイル時に組み込むように、コマンド行を使用して渡すことができます。
-K	probeVue セッションで RAS イベント関連機能を使用可能にします。

項目	説明
-l	<p>プローブ・マネージャーによってサポートされているすべてのプローブ・ポイントをリストします。probevue コマンドで -l フラグを指定する場合、使用しなければならないフラグは他にはありません。例えば -l syscall -l syscallx -l interval のように、-l フラグと一緒に複数のプローブ・マネージャーを指定することができます。</p> <p>プローブ・マネージャーは -l フラグに対して interval、syscall、syscallx、systrace、io、sysproc、および net プローブをサポートしています。-l オプションと一緒に誤った引数や不適当なプローブ・マネージャーを指定すると、使用法のエラーが表示されます。</p> <ul style="list-style-type: none"> • probevue -l syscall: システム上でトレースできるすべての可能なシステム・コールをリストします。 • probevue -l syscallx: システム上でトレースできるすべての基本システム・コールを表示します。このオプションは、32 ビットおよび 64 ビット・システムについて、別々にシステム・コールをリストします。 • probevue -l syscallx32: システム上でトレースできる 32 ビット基本システム・コールを表示します。 • probevue -l syscallx64: システム上でトレースできる 64 ビット基本システム・コールを表示します。 • probevue -l interval: interval プローブで通常のユーザーおよび root ユーザーについてサポートされている、最小および最大間隔の所要時間を指定します。 • probevue -l systrace: systrace プローブに関する記述を表示します。 注: syscallx プローブ・マネージャーの場合、-l syscallx プローブが使用されると、32 ビットと 64 ビットの両方の基本システム・コールを表示します。32 ビット・コールのみを表示するには syscallx32 プローブを使用し、64 ビット・コールのみを表示するには syscallx64 プローブを使用します。
-o Output_file	標準出力ではなくファイルにレポートを書き込みます。
-P C++ header file	<p>C++ ヘッダー・ファイルをプリプロセスし、入力された C++ ヘッダー・ファイルごとに出力されたプリプロセス済みファイルを作成します。プリプロセスされた出力ファイルの名前は、入力された C++ ヘッダー・ファイルと同じもので、.Vue 接尾部が付いています。 注: 他のフラグは、-P オプションと一緒に使用できません。-P フラグは、.Vue 接尾部の付いたファイル名以外のすべてのファイル名を受け入れます。</p>
-q info=level	<p>vue スクリプトの構文解析中の情報メッセージの報告レベルを指定します。有効な値は次のとおりです。</p> <p>none 情報メッセージは表示されません。</p> <p>normal 重要な情報メッセージのみが表示されます。</p> <p>detail 誤った実行結果になる可能性があるすべての情報メッセージが報告されます。</p>
-s Buffer_size	CPU 別のトレース・バッファ・サイズ (KB) を指定します。この値は次の 4K ページに繰り上げて丸めます。以下のレベルを使用できます。
-t Interval	<p>トレース・バッファを読み取る頻度を指定します。指定できる最小間隔は 10 ミリ秒です。通常のユーザー (aix.ras.probevue.trace 特権のないユーザー) が指定した時間間隔は、次に大きい 10 ミリ秒の倍数に丸められます。読み取り速度は probeVue 構成から取得されます。 注: 通常のユーザーは最小読み取り速度を指定できますが、probevuectrl コマンドはデフォルト読み取り速度を変更することができます。</p>
-T	セッションの開始時にプローブ・アクションのプロファイル作成を開始します。 -T フラグにより、セッションの開始時にプローブ・アクションのプロファイル作成が確実に行われるようになります。
-u	probeVue セッションを開始し、データ・セグメントを無制限の値に設定します。 -u フラグは、大規模なバイナリーの複数のプロセスからスタック・トレースを出力するために使用されます。
-X Program_name	プログラムを起動し、プログラムが実際に開始される前にプローブを使用可能にします。 vue スクリプトの中で特殊環境変数 \$_CPID と \$_CTID を使用して、起動されたアプリケーションのプロセス ID とスレッド ID を識別できます。

セキュリティ

RBAC ユーザーおよび **Trusted AIX ユーザー**への注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

1. スクリプト **syscall.e** を指定して **probeVue** セッションを開始するには、次のように入力します。

```
probevue syscall.e
```

2. トレース・レポートを **/tmp/trace_report** ファイルに送るには、次のように入力します。

```
probevue -o /tmp/trace_report syscall.e
```

3. スレッド ID 12345,4567 のトレース・レポートおよびトレースの開始時刻に相対的なタイム・スタンプを表示するには、次のように入力します。

```
probevue -c "timestamp=0 tid=12345,4567" syscall.e
```

4. ヘッダー・ファイル **stat.i** を組み込み、4K の CPU 別バッファを割り当てるには、次のように入力します。

```
probevue -I stat.i -s 4 syscall.e
```

5. C++ ヘッダー・ファイル **myheader.h** をプリプロセスするには、以下のように入力します。

```
probevue -P myheader.h
```

probevue コマンドは、**myheader.Vue** ファイルを生成します。このファイルは暗号化 C++ ヘッダー・ファイルであり、**-I** オプションを使用することによりトレース・セッションに組み込まれます。

6. 動的データ構造 (スタック・トレースおよび結合配列) の現行セッションについて **pinned** (滞留) されたメモリのパーセンテージを、**ASO.e** スクリプト用のデフォルトの 50 -75 から増やすには、以下のように入力します。

```
probevue -e 75 ASO.e
```

7. 以下のスクリプトは、スクリプト内で引数を提供する例です。

```
#!/usr/bin/probevue
#VUE_CMD_ARGS=-o /tmp/trace_out
#VUE_SCRIPT_ARGS=read

@@syscall::*:$1:entry
{
    printf("%t¥n", get_stktrace(4));
}
```

スクリプトは、**./script.e** として実行されます。

ファイル

項目	説明
<code>/usr/bin/probevue</code>	<code>probevue</code> コマンドが入っています。

関連資料:

460 ページの『`probevctrl` コマンド』

関連情報:

ProbeVue ユーザー・ガイド

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

項目	説明
<code>ProcessID</code>	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

項目	説明
<code>/proc</code>	<code>/proc</code> ファイルシステムが入っています。

関連資料:

- 『`procfiles` コマンド』
- 472 ページの『`procflags` コマンド』
- 474 ページの『`procldd` コマンド』
- 484 ページの『`proctree` コマンド』
- 488 ページの『`procwait` 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 ] [ -c ] 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 ASCIIi 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。

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

項目	説明
-c	列フォーマットで出力を表示します。
-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。
-n	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せん。
ProcessID	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、次のコマンドを入力します。

```
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、次のコマンドを入力します。

```
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
```

3. プロセス 278684 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、次のコマンドを入力します。

```
procfiles -c 278684
```

このコマンドの出力は以下のような情報です。

```
278684 : -ksh
Current rlimit: 2000 file descriptors
```

FD	TYPE	MODE	DEV/RDEV	UID	GID	OPMOD	INODE
0	c	-----	10, 4(19, 0)	root	system	R-W	16385
1	c	-----	10, 4(19, 0)	root	system	R-W	16385
2	c	-----	10, 4(19, 0)	root	system	R-W	16385
61	-	rw-r--r--	10, 7	root	system	R-W	32
63	-	rw-----	10, 4	root	system	R-W A	1051

ファイル

項目	説明
<code>/proc</code>	<code>/proc</code> ファイルシステムが入っています。

関連資料:

- 469 ページの『`proccred` コマンド』
- 『`procflags` コマンド』
- 474 ページの『`procldd` コマンド』
- 475 ページの『`procmmap` コマンド』
- 489 ページの『`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

項目	説明
-r	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。
ProcessID	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
gpr3  = 0x2ff22cb3
gpr4  = 0x11
gpr6  = 0x50
gpr7  = 0x0
gpr9  = 0x4c4f47
gpr10 = 0x80000000
gpr12 = 0x0
gpr13 = 0xdeadbeef
gpr15 = 0x2ff22c0c
gpr16 = 0x2ff22c14
gpr18 = 0xdeadbeef
gpr19 = 0xdeadbeef
gpr21 = 0xdeadbeef
gpr22 = 0x10
gpr24 = 0x2f
gpr25 = 0x2ff227f0
gpr27 = 0x2ff22d87
gpr28 = 0x2ff22cb3
gpr30 = 0x0
gpr31 = 0xf0048260
msr   = 0x2d032
ctr   = 0xec
xer   = 0x0
fpscr = 0x0
gpr2  = 0xf0083bec
gpr5  = 0x65
gpr8  = 0x41707a7c
gpr11 = 0x34e0
gpr14 = 0x1
gpr17 = 0x0
gpr20 = 0xdeadbeef
gpr23 = 0xfd
gpr26 = 0x0
gpr29 = 0x0
iar   = 0xd01be900
lr    = 0xd01d9de0
fpscr = 0x0
```

ファイル

項目	説明
/proc	/proc ファイルシステムが入っています。

関連資料:

- 469 ページの『[proccred コマンド](#)』
- 470 ページの『[procfiles コマンド](#)』
- 474 ページの『[procldd コマンド](#)』
- 483 ページの『[procstop コマンド](#)』
- 489 ページの『[procwdx 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用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用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

項目	説明
-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。
<i>ProcessID</i>	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

項目	説明
<code>/proc</code>	<code>/proc</code> ファイルシステムが入っています。

関連資料:

- 469 ページの『`proccred` コマンド』
- 472 ページの『`procflags` コマンド』
- 『`procmap` コマンド』
- 480 ページの『`procsig` コマンド』
- 489 ページの『`procwdx` コマンド』

procmap 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

```
procmap [ -F ] [ -S ] { -X [-f] [-n] [-u] [-q] } 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。

`procmap` 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/map` ファイルから収集されます。

フラグ

項目	説明
<code>-F</code>	別のプロセスが制御を持っていたとしても、強制的に <code>procmap</code> コマンドがターゲット・プロセスの制御を得るようにします。
<code>-S</code>	ターゲット・プロセスの共有メモリ情報を表示します。 注: <code>-S</code> オプションは廃止されました。 <code>-X</code> オプションを使用して同様の機能を実行できます。
<code>-X</code>	各アドレス範囲に関する拡張情報を表示し、ターゲット・プロセスのすべてのマップ・オブジェクトを含めます。
<code>-n</code>	ターゲット・プロセスのマップされたセグメントの未ソートのアドレス範囲を表示します。

項目	説明
-u	最も適切な単位で値を表示します。レポートに使用される単位は、メトリックごとに指定されます。
-q	ヘッダー情報を抑止します。
-f	ターゲット・プロセスの装置番号と inode 番号の代わりに、ファイル名を表示します。マップ・ファイルの「 <i>MAPPED OBJECT</i> 」列には、次のいずれかの値が表示されます。 <ul style="list-style-type: none"> • dev:remote リモート・ファイルの場合。 • dev:remote グローバル・ホスト・ストレージを使用して WPAR 内から。 • <full pathname> 直接ストレージを使用して WPAR 内から。 • dev:<major#>、<minor#>、ino:<inode#> 直接ストレージを使用して WPAR の外部に。 • 非 WPAR ローカル・ファイルの場合、ファイル名は解決されるか、またはストリング dev:<major#>、<minor#>、ino:<inode#> が、ファイルの正しいメジャー番号、マイナー番号、および inode 番号とともに表示されます。
ProcessID	プロセス ID を指定します。

以下は、-X フラグを持つ列の内容の簡単な説明です。

項目	説明
Start-ADD	メモリー領域の開始アドレス。
End-ADD	メモリー領域の終了アドレス。
SIZE	メモリー領域のサイズ。
MODE	メモリー領域の読み取り、書き込み、および実行の権限。
PSIZ	メモリー領域のページ・サイズ。以下の値が表示されます。
s	4K ページ・サイズ
m	64K ページ・サイズ
L	16M ページ・サイズ
H	16GB ページ・サイズ
TYPE	メモリー領域のタイプ。以下の値が表示されます。
HEAP	ヒープ領域
KER/LDR	その他のカーネル/ローダー・セグメント
KERTXT	カーネル・テキスト
MAINDATA	プロセスのメイン・データ/BSS
MAINTEXT	プロセスのメイン実行可能ファイル
MFILE	マップ・ファイル
PLIBDATA	専用ライブラリー・データ
PLIBTEXT	専用ライブラリー・テキスト
PMMAP	専用メモリー・マップ
PMMFILE	専用メモリー・マップ・ファイル
PMMPSXRT	専用 POSIX リアルタイム共有メモリー・マップ
REALMEM	マップされた実メモリー
SHM	共有メモリー
SHMEXT	拡張共有メモリー
SHMFILE	共有メモリー・ファイル
SLIBDATA	共有ライブラリー・データ
SLIBTEXT	共有ライブラリー・テキスト
SMMAP	共有メモリー・マップ
SMMFILE	共有メモリー・マップ・ファイル
SMMPSXRT	共有 POSIX リアルタイム共有メモリー
STACK	スタック領域
VSID	メモリー領域の仮想セグメント ID
MAPPED OBJECT	マップ・オブジェクト名

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。

```
procmap 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.o]
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
```

2. PID 1573580 を持つプロセスのアドレス範囲を表示するには、次のように入力します。

```
# procmap -X 1573580
1573580 : ./self_mod_32
```

Start-ADD	End-ADD	SIZE	MODE	PSIZ	TYPE	VSID	MAPPED OBJECT
0	10000000	262144K	r--	m	KERTXT	20002	
10000000	10001d33	7K	rwX	sm	MAINTEXT	1730DF3	self_mod_32
200001d0	200007fc	1K	rw-	sm	MAINDATA	4F0CCF	self_mod_32
200007fc	20011000	66K	rw-	sm	HEAP	4F0CCF	
20011000	20011685	1K	rwX	sm	PLIBTEXT	4F0CCF	./libself_priv.a[d1_prv32.o]
20011685	2001234c	3K	rw-	sm	HEAP	4F0CCF	
2001234c	200125c4	0K	rw-	sm	PLIBDATA	4F0CCF	./libself_priv.a[d1_prv32.o]
200125c4	20013000	2K	rw-	sm	HEAP	4F0CCF	
20013000	2ff23000	261184K	rw-	sm	STACK	4F0CCF	
30000000	30001000	4K	rw-	sm	SMMFILE	8C0C0C	dev:10,7 ino:35
30001000	30002000	4K	r--	sm	SMMFILE	8C0C0C	dev:10,7 ino:36
30002000	30003000	4K	rw-	sm	PMMFILE	8C0C0C	dev:10,7 ino:35
30003000	30004000	4K	r--	sm	PMMFILE	8C0C0C	dev:10,7 ino:36
30004000	30005000	4K	rw-	sm	SMMPXRT	8C0C0C	POSIX RT SHM 1
30005000	30006000	4K	r--	sm	SMMPXRT	8C0C0C	POSIX RT SHM 1
30006000	30007000	4K	rw-	sm	PMMPXRT	8C0C0C	POSIX RT SHM 2
30007000	30008000	4K	r--	sm	PMMPXRT	8C0C0C	POSIX RT SHM 2
30008000	30009000	4K	rw-	sm	SMMAP	8C0C0C	
30009000	3000a000	4K	r--	sm	SMMAP	8C0C0C	
3000a000	3000b000	4K	rw-	sm	PMMAP	8C0C0C	
3000b000	3000c000	4K	r--	sm	PMMAP	8C0C0C	
40000000	40000000	0K	rw-	s	SHMFILE	250CA5	dev:10,7 ino:35
50000000	50000000	0K	r--	s	SHMFILE	1960D16	dev:10,7 ino:36
60000000	60001000	4K	rw-	sm	SHM	D0C8D	shmid:16
70000000	70001000	4K	r--	sm	SHM	11F0D9F	shmid:17
d0100100	d052343c	4236K	rwX	m	SLIBTEXT	3010B81	/usr/lib/libc.a[shr.o]
d0564100	d0564abe	2K	rwX	m	SLIBTEXT	3010B81	/usr/lib/libcrypt.a[shr.o]
d0652100	d0653654	5K	rwX	m	SLIBTEXT	3010B81	./libself.a[support32.o]

```

d0654380      d0654a02          1K rwx  m   SLIBTEXT  3010B81      ./libself.a[shr32.o]
d0655a80      d0656105          1K rwx  m   SLIBTEXT  3010B81      ./libself.a[d1_shr32.o]
f05935cc      f0593844          0K rw-  sm   PLIBDATA  17F0DFF      ./libself.a[shr32.o]
f06a5b6f      f06a60c0          1K rw-  sm   PLIBDATA  17F0DFF      ./libself.a[support32.o]
f07b4ccc      f07b4f44          0K rw-  sm   PLIBDATA  17F0DFF      ./libself.a[d1_shr32.o]
f07dfbb0      f08b7388          861K rw- sm   PLIBDATA  17F0DFF      /usr/lib/libc.a[shr.o]
f08b86a8      f08b87c8          0K rw-  sm   PLIBDATA  17F0DFF      /usr/lib/libcrypt.a[shr.o]

```

Total 528579K

3. PID 2031848 を持つプロセスのマップ・ファイルに対するファイル名アソシエーションを持つアドレス範囲を表示するには、次のように入力します。

```

# procmap -Xf 2031848
2031848 : ./self_mod_64

```

Start-ADD	End-ADD	SIZE	MODE	PSIZ	TYPE	VSID	MAPPED OBJECT
0	10000000	262144K	r--	m	KERTXT	20002	
100000000	100002058	8K	rwx	sm	MAINTEXT	3C0CBC	self_mod_64
1100002d0	1100009e0	1K	rw-	sm	MAINDATA	240CA4	self_mod_64
1100009e0	110010a00	64K	rw-	sm	HEAP	240CA4	
800000000000e80	8000000000012eb	1K	rwx	sm	PLIBTEXT	5A0CDA	./libself_priv.a[d1_prv64.o]
8001000a0000180	8001000a00001c0	0K	rw-	sm	PLIBDATA	16A0DEA	./libself_priv.a[d1_prv64.o]
800200140000000	80020014003d000	244K	r--	sm	KER/LDR	2A20E22	
8ffffff00000000	900000000000000	262144K	r--	s	KER/LDR	530CD3	
900000000000e00	900000000440541	4349K	rwx	m	SLIBTEXT	28C0E0C	/usr/lib/libc.a[shr_64.o]
900000000466400	900000000466f43	2K	rwx	m	SLIBTEXT	28C0E0C	/usr/lib/libcrypt.a[shr_64.o]
900000000467980	900000000468e98	5K	rwx	m	SLIBTEXT	28C0E0C	./libself.a[support64.o]
900000000469100	900000000469568	1K	rwx	m	SLIBTEXT	28C0E0C	./libself.a[shr64.o]
90000000046a800	90000000046ac6b	1K	rwx	m	SLIBTEXT	28C0E0C	./libself.a[d1_shr64.o]
9001000a0000580	9001000a010cb88	1073K	rw-	sm	PLIBDATA	12A0DAA	/usr/lib/libc.a[shr_64.o]
9001000a010da28	9001000a010dbb8	0K	rw-	sm	PLIBDATA	12A0DAA	/usr/lib/libcrypt.a[shr_64.o]
9001000a0137380	9001000a01378b8	1K	rw-	sm	PLIBDATA	12A0DAA	./libself.a[support64.o]
9001000a0248400	9001000a0248440	0K	rw-	sm	PLIBDATA	12A0DAA	./libself.a[shr64.o]
9001000a045ab00	9001000a045ab40	0K	rw-	sm	PLIBDATA	12A0DAA	./libself.a[d1_shr64.o]
900200140000000	900200150000000	262144K	r--	s	KER/LDR	B70037	
9ffffffd0000000	9ffffffe0000000	262144K	r--	sm	KER/LDR	50005	
9ffffffe0000000	9ffffffe0000000	262144K	r--	sm	KER/LDR	E000E	
9ffffff00000000	9ffffff000fa8e	62K	rwx	s	SLIBTEXT	1180198	/usr/ccs/bin/us1a64
9ffffff000fa8e	9ffffff000fa8e	0K	rw-	s	PLIBDATA	1180198	/usr/ccs/bin/us1a64
a00000000000000	a00000000001000	4K	rw-	sm	SMMFILE	420CC2	/tmp/mmfile1
a00000000001000	a00000000002000	4K	r--	sm	SMMFILE	420CC2	/tmp/mmfile2
a00000000002000	a00000000003000	4K	rw-	sm	PMMFILE	420CC2	/tmp/mmfile1
a00000000003000	a00000000004000	4K	r--	sm	PMMFILE	420CC2	/tmp/mmfile2
a00000000004000	a00000000005000	4K	rw-	sm	SMMPSXRT	420CC2	POSIX RT SHM 1
a00000000005000	a00000000006000	4K	r--	sm	SMMPSXRT	420CC2	POSIX RT SHM 1
a00000000006000	a00000000007000	4K	rw-	sm	PMMPSXRT	420CC2	POSIX RT SHM 2


```

a00000000007000 a00000000008000 4K r-- sm PMMPSXRT 420CC2 POSIX RT SHM 2
a00000000008000 a00000000009000 4K rw- sm SMMAP 420CC2
a00000000009000 a0000000000a000 4K r-- sm SMMAP 420CC2
a0000000000a000 a0000000000b000 4K rw- sm PMMAP 420CC2
a0000000000b000 a0000000000c000 4K r-- sm PMMAP 420CC2
a00010000000000 a00010000000000 0K rw- s SHMFILE 250CA5 /tmp/mmfile1
a00010010000000 a000100100000000 0K r-- s SHMFILE 1960D16 /tmp/mmfile2
a00020000000000 a00020000001000 4K rw- sm SHM 5B0CDB shmid:18
a00030000000000 a00030000001000 4K r-- sm SHM 1980D18 shmid:19
ffffff000000000 1000000000000000 4194304K rw- sm STACK 10D0D8D
Total 5510897K

```

ファイル

項目	説明
<code>/proc</code>	<code>/proc</code> ファイルシステムが入っています。

関連資料:

- 469 ページの『`proccred` コマンド』
- 470 ページの『`procfiles` コマンド』
- 『`procrun` コマンド』
- 480 ページの『`procsig` コマンド』
- 488 ページの『`procwait` 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 `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。

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

項目	説明
<i>ProcessID</i>	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

項目	説明
<i>/proc</i>	<i>/proc</i> ファイルシステムが入っています。

関連資料:

- 469 ページの『**proccred** コマンド』
- 470 ページの『**procfiles** コマンド』
- 474 ページの『**procldd** コマンド』
- 475 ページの『**procmmap** コマンド』
- 『**procsig** 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

項目
ProcessID

説明
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

項目	説明
<code>/proc</code>	<code>/proc</code> ファイルシステムが入っています。

関連資料:

- 469 ページの『`proccred` コマンド』
- 470 ページの『`procfiles` コマンド』
- 『`procstack` コマンド』
- 483 ページの『`procstop` コマンド』
- 484 ページの『`proctree` コマンド』
- 488 ページの『`proccwait` 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 ] [ -g ] 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` コマンドはそれぞれ、指定されたプロセスについて `/proc` から情報を収集し、ユーザーに示します。`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

項目	説明
<code>-F</code>	別のプロセスが制御を持っていたとしても、強制的に <code>procstack</code> コマンドがターゲット・プロセスの制御を得るようにします。
<code>-g</code>	シンボル名が人間が理解できる名前に変換されないようにします。
<code>ProcessID</code>	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
10007alc job_wait  (?) + 144
10020298 xec_switch  (? , ? , ? , ? , ?) + 9c0
10021db4 sh_exec  (? , ? , ?) + 304
10001370 exfile  () + 628
10000300 main  (? , ?) + alc
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 nsleep  (? , ?) + b4
d026a6c0 sleep  (?) + 34
10000480 PrintHello  (c) + 30
d004b314 _pthread_body  (?) + ec
```

ファイル

項目	説明
<code>/proc</code>	<code>/proc</code> ファイルシステムが入っています。

関連資料:

- 469 ページの『`procrcd` コマンド』
- 470 ページの『`profiles` コマンド』
- 472 ページの『`proclags` コマンド』
- 474 ページの『`procldd` コマンド』
- 475 ページの『`procmmap` 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

項目	説明
<i>ProcessID</i>	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

項目	説明
/proc	/proc ファイルシステムが入っています。

関連資料:

- 469 ページの『**proccred** コマンド』
- 470 ページの『**procfiles** コマンド』
- 472 ページの『**procflags** コマンド』
- 480 ページの『**procsig** コマンド』
- 482 ページの『**procstack** 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 } ]
```

```
proctree [ -a ] [ -T ] [ -t ] [ { -p ProcessID | -u User } ] [ -@ [WparName] ]
```

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。

パラメーターのない **-@** フラグを指定すると、すべての WPAR 名が表示されます。 *WparName* パラメーターを指定した場合は、該当する WPAR 名だけが表示されます。

スレッド ID と関連の **pthread** ID を表示するには、**-t** オプションを指定します。カーネル・プロセスの場合、**proctree** コマンドはスレッド ID のみを表示します。

注: **proctree** コマンドは、プロセスに関する情報を取り出せない場合、プロセス ID のみを表示します。プロセスに関するその他の情報は、ブランクとして表示されます。例えば、**proctree** コマンドは、ゾンビ・プロセスのプロセス ID のみを表示します。

フラグ

項目	説明
-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。
<i>ProcessID</i>	プロセス ID を指定します。
-p ProcessID	プロセス ID を指定します。
-T	プロセス・ツリーの定様式出力を表示します。
-t	このプロセス用にスレッド ID と関連の pthread ID を表示します。
<i>User</i>	ユーザー名を指定します。
-u User	ユーザー名を指定します。
-@	すべての WPAR 名を表示します。 注: -@ フラグは、workload partition内で実行時はサポート対象外です。
-@ WparName	<i>WparName</i> パラメーターで指定した WPAR のプロセスだけが表示されます。 注: -@ フラグは、workload partition内で実行時はサポート対象外です。

例

1. プロセス 12312 の祖先とすべての子を表示するには、次のコマンドを入力します。

```
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、プロセス 0 の子を含めて表示するには、次のコマンドを入力します。

```
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
```

3. WPAR corral2 のプロセス・ツリーを表示するには、次のコマンドを入力します。

```
proctree -@ corral2
```

このコマンドの出力は以下のような情報です。

```
corral2 401496 /etc/init
corral2 319680 /usr/sbin/srcmstr
corral2 102636 /usr/sbin/inetd
corral2 249954 /opt/rsct/bin/rmcd -a IBM.LPCCommands -r
corral2 254132 /opt/rsct/bin/IBM.AuditRMd
corral2 295098 /opt/rsct/bin/IBM.ServiceRMd
corral2 303218 /usr/dt/bin/dtlogin
corral2 307370 /usr/sbin/writesrv
corral2 323836 /usr/sbin/qdaemon
corral2 331970 /usr/sbin/muxatmd
corral2 348210 /usr/sbin/syslogd
corral2 352472 sendmail: accepting connections H nnections
corral2 364564 /opt/rsct/bin/IBM.ERrmd
corral2 405522 /usr/sbin/portmap
corral2 282800 /usr/bin/xmwl -L
corral2 311454 /usr/sbin/cron
corral2 376920 /usr/lib/errdemon
```

4. プロセスの WPAR 名を表示するには、次のコマンドを入力します。

```
proctree -@
```

このコマンドの出力は以下のような情報です。

```
Global 114788 /usr/dt/bin/dtlogin -daemon
Global 86108 dtlogin <:0> -daemon
Global 123022 dtgreet 8 :0
Global 77944 /usr/lib/errdemon
Global 94314 /usr/sbin/syncd 60
Global 168084 /usr/sbin/srcmstr
Global 110688 /opt/rsct/bin/IBM.ServiceRMd
corral2 401496 /etc/init
corral2 319680 /usr/sbin/srcmstr
corral2 102636 /usr/sbin/inetd
corral2 249954 /opt/rsct/bin/rmcd -a IBM.LPCCommands -r
corral2 254132 /opt/rsct/bin/IBM.AuditRMd
corral2 331970 /usr/sbin/muxatmd
```



```

corral2  348210  /usr/sbin/syslogd
corral2  364564  /opt/rsct/bin/IBM.ERrmd
corral2  405522  /usr/sbin/portmap
corral2  282800  /usr/bin/xmwm -L
corral2  311454  /usr/sbin/cron
corral2  376920  /usr/lib/errdemon
Global  151626  /usr/ccs/bin/shlap64
Global  274578  /usr/sbin/getty /dev/console
...

```

5. プロセス 102636 の祖先、すべての子、および WPAR 名を表示するには、次のコマンドを入力します。

```
proctree -p 102636 -e
```

このコマンドの出力は以下のような情報です。

```

Global  168084  /usr/sbin/srcmstr
corral2  401496  /etc/init
corral2  319680  /usr/sbin/srcmstr
corral2  102636  /usr/sbin/inetd

```

6. プロセス 213246 のプロセス・ツリー定様式出力を表示するには、次のコマンドを入力します。

```
proctree -T -p 213246
```

このコマンドの出力は以下のような情報です。

```

192652      ¥--/usr/sbin/srcmstr
200830          ¥--/usr/sbin/inetd
213246              ¥--telnetd -a
229592                  ¥---ksh

```

7. 344172 プロセス用にスレッド ID と関連の pthread ID を表示するには、以下のコマンドを入力します。

```
proctree -t -p 344172
```

このコマンドの出力は以下のような情報です。

```

192652  /usr/sbin/srcmstr
      TID : 225535 (pTID :    1)
  200830  /usr/sbin/inetd
      TID : 360677 (pTID :    1)
  323642  telnetd -a
      TID : 770057 (pTID :    1)
  307428  -ksh
      TID : 1056861 (pTID :    1)
    344172  apthd
      TID : 1065119 (pTID :    1)
      TID : 1028171 (pTID :   258)
      TID : 1011789 (pTID :  2057)
      TID : 1024105 (pTID :  1800)

```

8. プロセス 344172 のプロセス・ツリー定様式出力を、スレッド ID と関連の pthread ID と一緒に表示するには、次のコマンドを入力します。

```
proctree -tT -p 344172
```

このコマンドの出力は以下のような情報です。

```

192652  ¥--/usr/sbin/srcmstr
      ^^TID : 225535 (pTID :    1)
  200830  ¥--/usr/sbin/inetd
      ^^TID : 360677 (pTID :    1)
  323642  ¥--telnetd -a
      ^^TID : 770057 (pTID :    1)
  307428  ¥---ksh
      ^^TID : 1056861 (pTID :    1)

```

```

344172          ¥--apthd
                ~TID : 1065119 (pTID :    1)
                ~TID : 1028171 (pTID :   258)
                ~TID : 1011789 (pTID :  2057)
                ~TID : 1024105 (pTID :  1800)

```

9. pconsole ユーザーのプロセス・ツリー定様式出力を表示するには、次のコマンドを入力します。

```
proctree -T -u pconsole
```

このコマンドの出力は以下のような情報です。

```

192652 ¥--/usr/sbin/srcmstr
225400   ¥--/bin/ksh /pconsole/lwi/bin/lwistart_src.sh
241824   ¥--/bin/ksh /pconsole/lwi/bin/lwistart_src.sh
340154   ¥--/usr/java5/bin/java -Xmx512m -Xms20m -Xscmx10m -Xshareclasses -Dfile.encoding=U

```

ファイル

項目	説明
/proc	/proc ファイルシステムが入っています。

関連資料:

- 469 ページの『[proccred コマンド](#)』
- 470 ページの『[procfiles コマンド](#)』
- 480 ページの『[procsig コマンド](#)』
- 483 ページの『[procstop コマンド](#)』
- 489 ページの『[procwdx 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。**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。

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

項目	説明
<code>-v</code> <code>ProcessID</code>	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。

```
procwait -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

項目	説明
<code>/proc</code>	<code>/proc</code> ファイルシステムが入っています。

関連資料:

- 469 ページの『`proccred` コマンド』
- 482 ページの『`procstack` コマンド』
- 483 ページの『`procstop` コマンド』
- 484 ページの『`proctree` コマンド』
- 『`procwdx` コマンド』

`procwdx` 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

```
procwdx [ -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。

procdx 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

項目	説明
-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。
<i>ProcessID</i>	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。

```
procdx 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

項目	説明
/proc	/proc ファイルシステムが入っています。

関連資料:

469 ページの『**proccred** コマンド』

480 ページの『**procsig** コマンド』

482 ページの『**procstack** コマンド』

483 ページの『**procstop** コマンド』

488 ページの『**procdwait** 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.gl** 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.gl** 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。

項目	説明
-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・ライン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。

項目	説明
-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・ライン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。

項目	説明
-g	非グローバル記号 (静的関数) を含めます。
-h	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。
-L PathName	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。
-m MonitorData	mon.out 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。
-s	mon.sum 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。
-S	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。
-v	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。
-z	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

項目	説明
mon.out	デフォルトのプロファイル。
a.out	デフォルトのオブジェクト・ファイル。
mon.sum	要約プロファイル。

関連資料:

179 ページの『**nm** コマンド』

関連情報:

gprof コマンド

commands コマンド

サブルーチンの概要

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

項目	説明
<i>nroffFlags</i>	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。
ファイル	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

項目	説明
-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".
-PPrinter	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

項目	説明
PRINTER	使用したい印刷キューを指定します。

ファイル

項目	説明
/usr/share/lib/nterm/tab.ppd	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。

関連資料:

605 ページの『**qprt** コマンド』

関連情報:

col コマンド
eqn コマンド
tbl コマンド

projctl コマンド

目的

プロジェクト・ベースの拡張アカウントリング・アクティビティーをサポートします。

構文

```
projctl add projname projnumber [comment] [ { -d projpath | -p [DN] } ]
```

```
projctl merge sourceprojpath [ -d targetprojfile ]
```

```
projctl rm projname [ { -d projpath | -p [DN] } ]
```

```
projctl chg projname [ -p pid [, pid] ] [-f]
```

```
projctl exec projname <cmd line> [-f]
```

```
projctl chattr agg projname {-s|-u} [ { -d projpath | -p [DN] } ]
```

```
projctl qpolicy [ -g [DN] ]
```

```
projctl qprojs [-n]
```

```
projctl qproj [projectname]
```

```
projctl qapp appname
```

```
projctl {chkusr | chkgrp | chkprojs | {{chkadm | chkall} [-d admpath]}}
```

```
projctl ldusr [ -r ] [ -a ]
```

```
projctl unldusr [ -a ]
```

```
projctl 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

項目	説明
-a	システム・リポ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

項目	説明
<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>	projectl exec 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>prospath</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。

projectl 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、**projectl** 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するには、**-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。

項目	説明
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 サブルーチンは、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必要があります。
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必要があります。

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・ライン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、**projctl 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。

項目	説明
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。

項目
ldadm

説明

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、別名エントリーおよびプロジェクト・エントリー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]

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ティング 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ティング 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。

項目	説明
ldprojs	システム・プロジェクト定義 <code>/etc/project/projdef</code> 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 (<code>/etc/project/projdef</code>) がカーネルにロードされます。
projctl ldprojs -g [DN] -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 <code>/etc/project/ldap/projdef</code> 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 [DN] -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・リブ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

エ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

さまざまなサブコマンドを、以下に示します。

項目	説明
unldadm	管理ポリシーをアンロードします。
unldall	すべてのロードされたポリシーをアンロードします。
unldgrp	グループ・ポリシーをアンロードします。
unldprojs	プロジェクト定義のみをアンロードします。
unldusr	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

項目	説明
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 chatr 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。

```
projctl 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。

```
projctl 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

項目	説明
<code>/usr/bin/projectl</code>	projectl コマンドが入っています。
<code>/etc/project/projdef</code>	システム・プロジェクト定義ファイルが入っています。
<code>/etc/project/ldap/projdef</code>	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。
<code>/etc/project/active</code>	現在ロードされたポリシーの状況が入っています。
<code>/etc/project/config</code>	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。
<code>/etc/security/ldap/ldap.cfg</code>	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。

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、**-noprepend** 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

項目	説明
-erase <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。
-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。
-kill <i>Character</i>	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。
-noprepend	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。
-norapid	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。
-prepend	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。
-rapid	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

項目	説明
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:	prompter 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。

セキュリティー

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティー」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**Issecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

ファイル

項目 説明
\$HOME/ html

関連情報:

comp コマンド

dist コマンド

whatnow コマンド

メール・アプリケーション

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。

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 **LC_COLLATE** 環境変数で使うことができます。

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

項目	説明
<code>/usr/sbin/proto</code>	<code>proto</code> 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

ファイルシステム

proxymngr コマンド

目的

プロキシ・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・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を始動し、さらに使用可能な全プロキシ・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・マネージャーは、できるだけ既存のプロキシを再利用しようとします。

プロキシ・マネージャーが取り扱うプロキシには、管理プロキシと非管理プロキシの 2 つのタイプがあります。

管理プロキシは、プロキシ・マネージャーの要求時に始動されるプロキシです。

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

項目	説明
<code>-config</code>	デフォルト <code>proxymngr config</code> ファイルをオーバーライドします。 <code>proxymngr config</code> ファイルの詳細は、以下を参照してください。
<code>-timeout</code>	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。
<code>-retries</code>	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。
<code>-verbose</code>	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・ライン・フラグを使って上書きできます。

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、非管理プロキシー・サービスまたは管理プロキシー・サービスを記述します。

非管理プロキシー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、非管理プロキシー・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 は非管理プロキシーのネットワーク・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。

管理プロキシーの場合、フォーマットは次のとおりです。

```
<service-name> managed <command-to-start-proxy>
```

service-name は、管理プロキシー・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、プロキシー・マネージャーによって、プロキシー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ーが始動できます。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>
!   または
!
!   <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・サービスの場合、プロキシ・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・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。非管理プロキシ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・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・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・マネージャーは非管理プロキシ・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** (円記号、文字 `t`) でタブ文字を、**¥n** (円記号、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 **¥** (円記号) 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
:R:	リリース番号	数値
:L:	レベル番号	数値
:B:	分岐番号	数値
:S:	シーケンス番号	数値
:I:	SCCS の ID 文字列 (SID)	:R::L::B::S:
:Dy:	デルタ作成年	YY
:Dm:	デルタ作成月	MM
:Dd:	デルタ作成日	DD
:D:	デルタ作成日付	YY/MM/DD
:Th:	デルタ作成時	HH
:Tm:	デルタ作成分	MM
:Ts:	デルタ作成秒	SS
:T:	デルタ作成時刻	HH/MM/SS
:DT:	デルタ・タイプ	D または R

項目	説明	値
:P:	デルタ作成者	ログイン名
:DS:	デルタ・シーケンス番号	数値
:DP:	直前のデルタ・シーケンス番号	数値
:Dt:	デルタ情報	:DT::I::D::T::P::DS::DP:
:Dn:	含まれるデルタのシーケンス番号	:DS: . . .
:Dx:	除外されるデルタのシーケンス番号	:DS: . . .
:Dg:	無視されるデルタのシーケンス番号	:DS: . . .
:DI:	含まれ、除外され、無視されるデルタのシーケンス番号	:Dn:/:Dx:/:Dg:
:Li:	デルタが挿入する行数	数値
:Ld:	デルタが削除する行数	数値
:Lu:	デルタが変更しない行数	数値
:DL:	デルタ行の統計情報	:Li:/:Ld:/:Lu:
:MR: (複数行フォーマット)	デルタの MR 番号	テキスト
:C: (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
:Y:	モジュール・タイプ	テキスト
:MF:	MR 妥当性検査フラグ・セット	yes または no
:MP:	MR 妥当性検査プログラム名	テキスト
:KF:	キーワード/エラー警告フラグのセット	yes または no
:BF:	分岐フラグのセット	yes または no
:J:	結合編集フラグのセット	yes または no
:LK:	ロック・リリース	:R: . . .
:Q:	ユーザー定義のキーワード	テキスト
:M:	モジュール名	テキスト
:FB:	フロア境界	:R:
:CB:	シーリング境界	:R:
:Ds:	デフォルト SID	:I:
:ND:	null デルタ・フラグ・セット	yes または no
:FL: (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に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
:UN: (複数行フォーマット)	ユーザー名	テキスト
:FD: (複数行フォーマット)	記述テキスト	テキスト
:BD: (複数行フォーマット)	テキストの本体	テキスト
:GB: (複数行フォーマット)	g ファイル内のテキスト	テキスト
:W:	what 文字列	:Z::M: ¥tab :I:
:A:	what 文字列	:Z::Y::M::I::Z:
:Z:	what 文字列の区切り文字	@(#)
:F:	SCCS ファイル名	テキスト
:PN:	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。

項目	説明
-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 (rmidel コマンドを参照してください)。
-c Cutoff	-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。 -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、スペースを入れしないでください。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。

項目	説明
0	正常終了。
>0	エラーが発生しました。

例

- 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 (**rmidel** 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
```

- 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

項目	説明
/usr/bin/prs	prs コマンドが入っています。

関連情報:

delta コマンド

sccsfile コマンド

ソース・コード制御システム (SCCS) の概要

prtacct コマンド

目的

acct 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

項目	説明
-f Fields	acctmerg 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。
-v	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。
-W	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。このフラグを使用すると、 prtacct コマンドが、 tacctx 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。 -W フラグと -X フラグを一緒に使用すると、 -X が優先されます。
-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。このフラグを使用すると、 prtacct コマンドが、 tacctx 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。 -W フラグと -X フラグを一緒に使用すると、 -X 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
```

接続時間アカウントリング・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トリング・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トリング・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

項目	説明
/usr/sbin/acct	アカウントティング・コマンドへのパス。
/var/adm/pacct	プロセスのアカウントティング用の現行ファイル。
/var/adm/pacct*	pacct 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 コマンド

acctdisk コマンド

acct コマンド

システム・アカウントティング

アカウントティング・サブシステムの設定

prtcnf コマンド

目的

システム構成情報を表示します。

構文

```
prtcnf [ -c ] [ -k ] [ -L ] [ -m ] [ -s ] [ -v ]
```

説明

フラグを指定せずに **prtcnf** 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

項目	説明
-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。

```
prtcnf
```

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       P1-01     Mouse Adapter
mouse0       P1-01-Lmouse3 button mouse
siota0       P1/Q1     Tablet Adapter

    Device Specific.(YL).....P1/Q1

pau0         P1/Q2     Ultimeidia Integrated Audio

    Device Specific.(YL).....P1/Q2

ppa0         P1/R1     CHRP 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

prtglbconfig コマンド

目的

prtglbconfig コマンドは、AIX 印刷サブシステム用のグローバル設定を構成します。

構文

```
prtglbconfig [ -s name = value ] [ -r name ]
```

説明

prtglbconfig コマンドは、印刷サブシステム設定を行うか、その設定をデフォルト値にリセットします。現在、このコマンドは ERRMSGCONTROL 設定を行うのに使用されます。この設定は、グローバル・プリンター・メッセージに影響します。この設定を使用して、以下のオプションのいずれかを選択できます。

- ALLON (すべてのメッセージがオンにされる)。
- LOGALL (すべてメッセージがオンにされるが、ログ・ファイルにログが書き込まれる)。
- CRITON (最も重大なエラー・メッセージだけがオンにされる)。
- ALLOFF (すべてのメッセージがオフにされる)。

現在、LOGALL オプションと CRITON オプションは ALLON オプションと同じです。

フラグ

項目	説明
-s <i>name</i> = <i>value</i>	<i>name</i> パラメーターで指定された印刷サブシステム設定は、 <i>value</i> パラメーターに指定された値を使って設定される旨を指定します。
-r <i>name</i>	<i>name</i> パラメーターに指定された印刷サブシステム設定をデフォルト値にリセットします。

セキュリティ

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

1. 印刷サブシステムにより生成されたすべてのメッセージを無視するように印刷サブシステムを設定するには、次のように入力します。

```
prtglbconfig -s ERRMSGCONTROL=ALLOFF
```

2. 印刷サブシステムのエラー・メッセージ制御をデフォルト値にリセットするには、次のように入力します。

```
prtglbconfig -r ERRMSGCONTROL
```

注: 印刷サブシステムにより生成された一部のメッセージは無視することができず、コンソール・ログ・ファイルに書き込まれます。印刷ジョブの開始方法の詳細は、**qprt** コマンドを参照してください。

ファイル

項目	説明
<code>/etc/prtglobalconfig</code>	グローバル構成ファイルが入っています。
<code>/usr/sbin/qdaemon</code>	qdaemon デーモンが入っています。
<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:

`/etc/prtglobalconfig` コマンド

`/etc/qconfig` 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 ] [ -@ [ WparName ] ]
```

Berkeley 標準

```
ps [ a ] [ c ] [ e ] [ ew ] [ eww ] [ ewww ] [ g ] [ n ] [ 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・ライン・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 コマンドの出力が workload partition (WPAR) 名を含んでいないが、現行稼働環境下のワークロード・パーティション内で実行するプロセスが関連する、プロジェクト ID (**PROJECT**)、ユーザー ID (**UID** あるいは **USER**)、またはグループ ID (**GID**) を含んでいる場合、ID にはプラス記号 (+) が前に付いて workload partition との関連を示します。各 workload partition には、ユーザー、グループ、およびプロジェクト ID のそれ自体の定義が含まれており、それらの ID は、グローバル環境用に定義したものとは異なっている可能性があります。**-@** オプションを指定して、出力の中に workload partition の名前を組み込むことができます。

注: **ps** コマンドは、アプリケーションがメモリーを解放した場合に、メモリー使用量カウンターの減少を表示しません。メモリーがアプリケーションから解放されると、そのメモリーは、プロセスごとのメモリー・フリー・リストに割り当てられます。**ps** コマンドは、解放されたメモリーを、アプリケーションに割り当てられたメモリーとして見なします。

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は括弧内に示されています。

ADDR

(**-l** および **1** 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**、**1**、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 **1** 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
SLOAD	0x00000001	プロセスがコア・メモリーで動作中であることを示す。
SNOSWAP	0x00000002	プロセスをスワップアウトできないことを示す。
STRC	0x00000008	プロセスがトレースされていることを示す。
SWTED	0x00000010	トレース中に、プロセスが停止したことを示す。
SFWTED	0x00000020	トレース中に、 fork 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。
SEWTED	0x00000040	トレース中に、 exec 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。
SLWTED	0x00000080	トレース中に、 load または unload 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。
SFIXPRI	0x00000100	pcpu 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。
SKPROC	0x00000200	カーネル・プロセスを示す。
SOMASK	0x00000400	シグナルを受信した後で、古いマスクの復元を示す。
SWAKEONSIG	0x00000800	シグナルが sleep サブルーチンを異常終了することを示す。その内容は、 PCATCH フラグの内容に等しいものであってはならない。 PCATCH および SWAKEONSIG の両方の内容が、 PMASK の内容よりも大きくなければならない。
SUSER	0x00001000	プロセスがユーザー・モードにあることを示す。
SLKDONE	0x00002000	プロセスがロックを行ったことを示す。
STRACING	0x00004000	プロセスがデバッグ・プロセスであることを示す。
SMPTRACE	0x00008000	マルチプロセス・デバッグを示す。
SEXIT	0x00010000	プロセスが終了していることを示す。
SSEL	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。
SORPHANPGRP	0x00040000	孤立プロセス・グループを示す。
SNOCNTLPROC	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。
SPPNOCLDSTOP	0x00100000	子プロセスが停止するときに親プロセスへ SIGHLD シグナルが送信されないことを示す。
SEXECED	0x00200000	プロセスが稼働したことを示す。
SJOBSESS	0x00400000	現行セッションでジョブ制御が使われたことを示す。
SJOB OFF	0x00800000	プロセスがジョブ制御を受けていないことを示す。
PSIGDELIVERY	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。
SRMSHM	0x02000000	exit 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。
SSLOTFREE	0x04000000	プロセスのロットが空いていることを示す。
SNOMSG	0x08000000	これ以上 uprintf サブルーチン・メッセージがないことを示す。

WPAR

(-@ フラグ) ワークロード区画名が入っています。 -@ フラグのを指定すると、**ps** コマンドは、プロセスが実行されているワークロード区画の名前を表示します。プロセス情報を表示するには、-@ フラグに *wparname* パラメーターを指定します。

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
SLOAD	0x00000001	プロセスがコア・メモリーで動作中であることを示す。
SNOSWAP	0x00000002	プロセスをスワップアウトできないことを示す。
STRC	0x00000008	プロセスがトレースされていることを示す。
SKPROC	0x00000200	カーネル・プロセスを示す。
SEXIT	0x00010000	プロセスが終了していることを示す。
SLPDATA	0x00020000	プロセスで大きなページが使用されることを示す。
SEXECED	0x00200000	プロセスが稼働したことを示す。
SEXECING	0x01000000	プロセスが実行中 (exec を実行中) であることを示す。
SPSEARLYALLOCC	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。
TKTHREAD	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 (-I および I 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, I、および -I フラグ) 親プロセスのプロセス ID。

PRI (-I および I 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。 **PROJECT** と **USER** フィールドは、現行稼働環境下の workload partition 内で実行中のプロセスに対する名前に変換されません。 **-U** と **-u** フラグだけは、現行稼働環境に適用されます。ただし、 **-@** フラグが特定のワークロード・パーティション名を指定して組み込まれている場合はその限りではありません。 **-@** フラグを使用して現行稼働環境以外のワークロード・パーティションを指定し、かつ、 **-U** と **-u** フラグが指定されている場合、ユーザー ID のリストは数値でなければなりません。

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 (-I および I 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。スケジューリング・ポリシーは、**sched** フラグが指定された場合のみ表示されます。

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 *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。

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。 **PROJECT** と **USER** フィールドは、現行稼働環境下の *workload partition* 内で実行中のプロセスに対する名前に変換されません。

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。
3. **ps** プログラムは、ローカル・リソースでユーザーおよびグループ情報を検索します。

フラグ

以下のフラグの前には、- (マイナス符号) が付きます。

項目	説明
-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。
-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 <i>Clist</i>	<i>Clist</i> 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。 <i>Clist</i> 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 <i>Format</i>	-o <i>Format</i> と同じ。
-f	完全なリストを生成します。
-G <i>Glist</i>	<i>Glist</i> 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。 <i>Glist</i> 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 <i>Glist</i>	<i>Glist</i> 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。 <i>Glist</i> 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。1 フラグも参照してください。
-L <i>pidlist</i>	<i>pidlist</i> 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。 <i>pidlist</i> 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 <i>Format</i> 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 <i>NameList</i>	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。

説明

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・ライン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が、すべてのコマンド・ライン引数が含まれます。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・ライン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。

項目
-o (続き)

説明	
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。
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。
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。
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。
scount	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 SC です。
sched	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 SCH です。
shmpgsz	プロセスの共有メモリー・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。
tag	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 TAG です。このタグは、最高 30 文字の長さの文字列で、 ps 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。
tcpu	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。このコマンドは WLM がアクティブまたはパッシブ・モードのいずれかで実行中に情報を表示し、それ以外の場合、このフィールドは - (ハイフン) として表示されます。このフィールドのデフォルト・ヘッダーは TCPU です。
tctime	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 TCTIME です。
tdiskio	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 TDISKIO です。
tpgsz	プロセスのテキスト・ページ・サイズを示します。
vmsize	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 VMSIZ が表示されます。 VMSIZ 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。

項目
-o (続き)

説明

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。

注: **ps -o THREAD** フラグは、スケジューラー・ポリシーを表示しません。スケジューリング・ポリシーは、**sched** フラグが指定された場合のみ表示されます。

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。

ユーザー

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 が表示されません。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。

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・ライン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。

項目

説明

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、コマンド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、コマンド 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。

項目	説明
-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	このフラグは -U Ulist フラグと同等です。 -u フラグだけは、現行稼働環境に適用されます。ただし、 -@ フラグも指定されている場合はその限りではありません。 -@ フラグを使用して現行稼働環境以外のワークロード・パーティションを指定し、かつ、 -u フラグが指定されている場合、ユーザー ID のリストは数値でなければなりません。
-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。 -U フラグだけは、現行稼働環境に適用されます。ただし、 -@ フラグも指定されている場合はその限りではありません。 -@ フラグを使用して現行稼働環境以外のワークロード・パーティションを指定し、かつ、 -U フラグが指定されている場合、ユーザー ID のリストは数値でなければなりません。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。
-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。
-@ [WparName]	ワークロード区画 WparName に関連付けられたプロセス情報を表示します。 WparName パラメーターを指定しない場合、すべてのワークロード区画のプロセス情報が表示されます。ワークロード区画情報はすべてのプロセスについて表示されます。どのプロセスの情報を表示するかを決めるには、 ps コマンドに他のフラグを指定する必要があります。

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。

項目	説明
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ラップし、 LINE_MAX 値に到達するまで ENV リストを表示します。
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 lft0 、端末 /dev/pts/2 の場合には pts/2 と指定してください。

項目	説明
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。
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。

終了状況

このコマンドは、以下の終了値を戻します。

項目	説明
0	正常終了。
>0	エラーが発生しました。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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
```

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
```

- 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
```

- /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
```

- 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=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
```

- 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  SHMPGSZ  CMD
41856 pts/15 0:00  4K   4K   4K   64K   ps
84516 pts/15 0:00  4K   4K   4K   64K   ksh
```

ファイル

項目	説明
<code>/usr/bin/ps</code>	<code>ps</code> コマンドが入っています。
<code>kill</code> コマンド、 <code>nice</code> 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
SLOAD	0x00000001	プロセスがコア・メモリーで動作中であることを示す。
SNOSWAP	0x00000002	プロセスをスワップアウトできないことを示す。
STRC	0x00000008	プロセスがトレースされていることを示す。
SKPROC	0x00000200	カーネル・プロセスを示す。
SEXIT	0x00010000	プロセスが終了していることを示す。
SEXECED	0x00200000	プロセスが稼働したことを示す。
SEXECING	0x01000000	プロセスが実行中 (exec を実行中) であることを示す。
TKTHREAD	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印刷される)。

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 単位で)。

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が、すべてのコマンド・ライン引数が含まれます。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・ライン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 プロセスの有効なグループ 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。

グループ

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。

ユーザー

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。

項目	説明
-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 <i>pgrplist</i>	<i>pgrplist</i> 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。 <i>pgrplist</i> の値は数字でなければなりません。
-G <i>grplist</i>	<i>grplist</i> 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 <i>format</i>	<i>format</i> で指定されたフォーマットで情報を表示します。 <i>format</i> 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ッションに記述されています。
-p <i>proclist</i>	<i>proclist</i> 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。 <i>proclist</i> 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 <i>sidlist</i>	<i>sidlist</i> 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 <i>termlist</i>	<i>termlist</i> 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 <i>uidlist</i>	<i>uidlist</i> 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 <i>uidlist</i>	<i>uidlist</i> 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	各ユーザー名および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。

項目	説明
-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。

項目	説明
0	正常終了。
>0	エラーが発生しました。

セキュリティ

RBAC ユーザーおよび **Trusted AIX ユーザー**への注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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
```

14. スレッドのスケジューリング・ポリシーを表示するには、以下のコマンドを入力します。

```
#ps -m -o THREAD,sched
USER  PID  PPID  TID ST  CP  PRI  SC  WCHAN  F  TT  BND  COMMAND  SCH
suresana 1609830 4227284  -  A  16  68  1  -  200001 pts/144  -  ps -m
-o THREAD sched 0
-  -  -  6381739 R  16  68  1  -400000  -  -  -  0
suresana 4227284 4239476  -  A  1  60  1  -200801 pts/144  -  bash  0
-  -  -  4177981 S  1  60  1  -410400  -  -  -  0
suresana 4239476 921694  -  A  0  60  1  -240001 pts/144  -  -ksh  0
-  -  -  5554385 S  0  60  1  -10400  -  -  -  0
```

ファイル

項目	説明
<code>/usr/sysv/bin/ps</code>	System V R4 <code>ps</code> コマンドが入っています。
<code>/etc/passwd</code>	ユーザー ID 情報が入っています。
<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。

関連資料:

86 ページの『`nice` コマンド』

関連情報:

`kill` コマンド

`ps` コマンドの使用法

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。

項目	説明
<code>-C</code>	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。
<code>-lLeft,Bottom</code>	変換済みラスター・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。 <code>Left</code> および <code>Bottom</code> 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。
<code>-m</code>	4014 の "Margin 2" モードを使用可能にします。
<code>-N</code>	次行の下と左マージン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。
<code>-pOutFile</code>	標準出力ではなくて <code>OutFile</code> 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。
<code>-R</code>	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。
<code>-sWidth,Height</code>	印刷ページの変換済みラスター・イメージのサイズを指定します。 <code>Width</code> および <code>Height</code> 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。

項目	説明
-SWidth	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力ラスター・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

html を参照してください。

関連情報:

NLSvec コマンド

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

項目	説明
<code>-fBodyfont</code>	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 <code>Courier</code> です。
<code>-pFile</code>	標準出力ではなくて <code>File</code> 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。
<code>-sPitch</code>	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 のいずれかとなります。デフォルトは 12 です。
<code>-FBoldfont</code>	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 <code>Courier-Bold</code> です。

各国文字サポート

`html` を参照してください。

関連資料:

230 ページの『`nroff` コマンド』

関連情報:

`enscript` コマンド

`NLSvec` コマンド

`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、標準入力が使われます。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/devpsc/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` (円記号、`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・ライン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
EntryName	enscript コマンドか psdit コマンドの -M オプションで提供されるメディア名に一致する文字列。
MediaWidth	ポイント単位のメディアの幅。
MediaDepth	ポイント単位のメディアの深さ。
ImageableLLX	ポイント単位のイメージ可能な左下隅の x 座標。
ImageableLLY	ポイント単位のイメージ可能な左下隅の y 座標。
ImageableURX	ポイント単位のイメージ可能な右下隅の x 座標。
ImageableURY	ポイント単位のイメージ可能な右下隅の y 座標。
PageRegionName	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。
PaperTrayName	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
Letter	Width 612 Depth 792 llx 18 lly 17 urx 597 ury 776 ページ-領域-名 Letter ページ-トレイ-名
Legal	Width 612 Depth 1008 llx 18 lly 17 urx 597 ury 992 ページ-領域-名 Legal ページ-トレイ-名

フラグ

-f1 *CodeSet:Font*

項目

-F*FontDirectory*

-M*MediaName*

-p*Prologue*

-o *List*

説明

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。

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。

コマンド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

項目	説明
PSLIBDIR	psc と psdit コマンド・プロローグ用 <code>/usr/lib/ps</code> ファイルの代わりに使用されるディレクトリーのパス名。
TRANSCRIPT	MBCS ハンドル用 <code>/usr/lib/ps/transcript.conf</code> の代わりに使用されるファイルの絶対パス名。

ファイル

項目	説明
<code>/usr/lib/font/devpspc/*</code>	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。
<code>/usr/lib/ps/psdit.pro</code>	デフォルトの PostScript プロローグが入っています。
<code>/usr/lib/ps/MediaSizes</code>	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。
<code>/usr/lib/ps/transcript.conf</code>	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

troff コマンド、PE コマンド、nroff および troff コマンド用 me マクロ・パッケージ

pshare コマンド

目的

共有ログイン・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、共有ポートを使用可能にします。共有ポートは、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有ポートの名前を報告します。共有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。

- `/dev/tty1` デバイスなど、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

項目	説明
<code>-a</code>	すべてのポートを共用として使用可能にします。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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
```

項目	説明
<code>/etc/inittab</code>	システムの初期化を制御します。

ファイル

項目	説明
<code>/etc/locks</code>	pshare および pdelay コマンドの lock ファイルが入っています。
<code>/usr/sbin/pshare</code>	pshare コマンドが入っています。

関連資料:

350 ページの『**penable** コマンド』

355 ページの『**phold** コマンド』

552 ページの『**pstart** コマンド』

関連情報:

getty コマンド

init 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

項目	説明
<code>-gPrologue</code>	デフォルトの PostScript プロローグの代わりに <i>Prologue</i> 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

html を参照してください。

関連資料:

540 ページの『ps4014 コマンド』

関連情報:

lpr コマンド

lp コマンド

NLSvec コマンド

psrasc コマンド

目的

集中 RAS データを収集します。

構文

```
psrasc type [ -d ] [ -n number ] -o outputFile logSpace/logStream
```

説明

psrasc コマンドは、PowerHA pureScale® ログ・ストリーム上に集められた信頼性・可用性・保守性 (RAS) データ・ログ・レコードを抽出し、RAS データ AIX フォーマットでファイルを作成します。PowerHA pureScale サービス名は **CentralizedLogService** です。このサービス名のバインディング情報は、**psrasc** コマンドを使用する前にセットアップする必要があります。

RAS データ型

指定された型が **syslog** の場合は、ログ・レコードには、メッセージ・イニシエーター・ホスト名を含むシステム・ログ・メッセージが含まれます。生成されたファイルのフォーマットは、システム・ログ宛先ファイルに類似しています。指定された型が **errlog** の場合は、ログ・レコードにはエラー・ログ・エントリーが含まれます。生成されたファイルは、あとで **errpt** コマンドによって活用できるエラー・ログ・ファイルです。

フラグ

項目	説明
タイプ	ログ・レコードに含まれる RAS データの型を指定します。このパラメーターは、最初に指定する必要があります。サポートされる RAS データ型は、 syslog と erlog です。この型に応じて、出力ファイルのフォーマットが決まります。
-d	収集されたログ・レコードが削除されることを指定します。
-n number	収集するログ・レコードの数を指定します。最も古いログ・レコードが収集されます。このパラメーターを指定しない場合は、すべてのログ・レコードが収集されます。
-o outputFile	出力ファイルの相対パス名または絶対パス名を指定します。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。
log_space/ log_stream	システム・ログ・メッセージの収集元のログ・ストリームのフルネームを指定します。フルネームは、/ (スラッシュ) で区切られた親ログ・スペース名とログ・ストリーム名で構成されます。

終了状況

このコマンドは、以下の終了値を戻します。

項目	説明
0	正常終了。
> 0	エラーが発生しました。

例

1. **CentralizedRAS/Syslog** という名前のログ・ストリームのログ・レコードを収集して、**CentralizedLogService** サービス名で識別される

PowerHA pureScale サーバー上の **syslog.out** ファイルに入れるには、次のように入力します。

```
psrasc syslog -o syslog.out CentralizedRAS/Syslog
```

2. **CentralizedRAS/Syslog** という名前のログ・ストリームの最も古い 100 個のログ・レコードを収集して、**CentralizedLogService** サービス名で識別される

PowerHA pureScale サーバー上の **/var/adm/ras/cluster_syslog** ファイルに入れ、さらにそれらのログ・レコードを削除するには、次のように入力します。

```
psrasc syslog -d -n 100 -o /var/adm/ras/cluster_syslog CentralizedRAS/Syslog
```

3. **CentralizedRAS/Errlog** という名前のログ・ストリームのログ・レコードを収集して、**CentralizedLogService** サービス名で識別される PowerHA pureScale サーバー上の **centralizedRAS_errlog** ファイルに入れるには、次のように入力します。

```
psrasc errlog -o centralizedRAS_errlog CentralizedRAS/Errlog
```

4. **CentralizedRAS/Errlog** という名前のログ・ストリームの最も古い 100 個のログ・レコードを収集して、**centralizedRAS_error** サービスで識別される

PowerHA pureScale サーバー上の **/var/adm/ras/cluster_errlog** エラー・ログ・ファイルに入れ、さらにそれらのログ・レコードを削除するには、次のように入力します。

```
psrasc errlog -d -n 100 -o /var/adm/ras/cluster_errlog CentralizedRAS/Errlog
```

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

項目	説明
-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。 <i>Pagespec</i> パラメーターはスペースなしの文字列です。 <i>Pagespec</i> パラメーターは、単一のページ番号または <i>N-M</i> フォーマットの範囲にすることができ、 <i>N</i> から <i>M</i> までのページを印刷します。 <i>-N</i> は、文書の最初からページ <i>N</i> までを印刷します。 <i>M-</i> は、ページ <i>M</i> 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

項目	説明
/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に使用する **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
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

項目	説明
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)	(none)
ZapfChancery-MediumItalic (ZC)	Zapf
ZapfDingbats	(none)

フラグ

項目	説明
-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	<i>Queue</i> 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	<i>Times</i> デフォルト・ファミリーではなく、 <i>R/I/B/BI</i> フォント用の指定フォント・ファミリーを使用します。 <i>Times</i> 、 <i>Courier</i> 、 <i>Helvetica</i> 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	<i>Number</i> 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	<i>Flag</i> 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	<i>PostScript</i> 出力をプリンターヘ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	最初のバナー・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す。

パラメーター

項目	説明
<code>troffFlags</code> ファイル	troff コマンドで使用可能な標準フラグを指定します。 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

項目	説明
<code>/usr/share/lib/tmac/tmac.*</code>	標準マクロ・ファイルが入っています。
<code>/usr/lib/font/devps/*</code>	PostScript 仮想デバイスの troff 記述ファイルが入っています。
<code>/usr/lib/ps/*.afm</code>	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。
<code>/usr/lib/ps/font.map</code>	フォント名とその省略形のリストが入っています。
<code>/usr/lib/ps/ditroff.font</code>	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
`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。**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ーを読み込んだとき、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

項目	説明
-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。

セキュリティー

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティー」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
/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/pstart	pstart コマンド・ファイルが入っています。

関連資料:

341 ページの『**pdelay** コマンド』

355 ページの『**phold** コマンド』

545 ページの『**pshare** コマンド』

関連情報:

init コマンド

inittab コマンド

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

項目	説明
-a	プロセス・テーブル内のエントリーを表示します。
-A	カーネル・スレッド・テーブル内のエントリーをすべて表示します。
-f	ファイル・テーブルを表示します。
-i	i ノード・テーブルと i ノード・データ・ブロックのアドレスを表示します。
-p	プロセス・テーブルを表示します。
-P	実行可能なカーネル・スレッド・テーブル・エントリーのみを表示します。
-s	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。
-S	プロセッサの状況を表示します。
-t	tty 構造体を表示します。
-u ProcSlot	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。
-T	システム変数を表示します。これらの変数は、var.h に簡単に説明されています。
-U ThreadSlot	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。

セキュリティ

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**Issecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
<code>/usr/sbin/pstat</code>	pstat コマンドが入っています。
<code>/dev/mem</code>	デフォルトのシステム・イメージ・ファイル。
<code>/usr/lib/boot/unix</code>	デフォルトのカーネル・イメージ・ファイル。
<code>/usr/include/sys/*.h</code>	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。

関連資料:

182 ページの『nmon コマンド』

関連情報:

stty コマンド

ptpd デーモン

目的

Precision Time Protocol (1588-2008) デーモン (**ptpd**) を開始します。

構文

```
/usr/sbin/ptpd [ -? ] [ -h ] [ -H ] [ -e setting ] [ -k ] [ -v ] [ -O ] [ -L ] [ -A ] [ -s ] [ -m ] [ -M ]
[ -y ] [ -E ] [ -P ] [ -a ] [ -n ] [ -C ] [ -V ] [ -c file ] [ -R dir ] [ -f file ] [ -S file ] [ -d
domain_number ] [ -u IP_address ] [ -r number ] [ -l file ] [ -i dev ]
```

説明

ptpd デーモンは、IEEE 1588-2008 標準で定義された Precision Time Protocol (PTP) バージョン 2 を実装します。PTP は、LAN で接続されたコンピューターの正確な時間調整を行います。このデーモンは、システム・クロックを操作し、下の方のポート番号を使用するために、root 権限を使用して実行する必要があります。**ptpd** デーモンは、IPv4 マルチキャスト、ユニキャスト、ハイブリッド・モード (混合)、およびイーサネット・モードの操作をサポートします。**ptpd** デーモンは、ハードウェアの支援がなくても、マイクロ秒以下のレベルのタイミング精度を達成し維持できます。

`/etc/ptpd2.conf` 構成ファイル (デフォルト・ファイル) を使用して、**ptpd** デーモンを構成します。ショート (**-x**) フラグとロング (**--xxxxx**) フラグは、デーモンの操作に対する基本制御を提供し、基本 PTP プロトコル設定のみを提供します。その他の設定は、**-h**、**-H**、および **-e** の各フラグで表示できます。

ptpd デーモンは、システム・リソース・コントローラー (SRC) から、コマンド・ラインからも開始できます。

ptpd デーモンを操作するには、次の SRC コマンドを使用します。

startsrc

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。

stopsrc

サブシステム、サブシステムのグループ、またはサブサーバーを終了します。

refresh

サブシステム、またはサブシステムのグループに、再度、適切な構成ファイルを読み取らせませす。

lssrc

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。

注: **ptpd** デーモンは、Network Time Protocol (NTP) および Simple Network Management Protocol (SNMP) 関連の機能をサポートしません。

ptpd デーモンでは、以下の用語を使用します。

スレーブ・デバイス

ptpd デーモンを実行しているシステム。 *master* デバイスからのコマンドを受け入れ、システム時刻を同期化して、関連付けられた境界クロック時間に一致させます。

マスター・デバイス

境界クロックに結合されているデバイス。同じネットワークにある 1 つ以上の PTP スレーブ・デバイスのセットと時刻を同期化します。

グランドマスター・デバイス

Best Master Clock アルゴリズムにより提供された最良のクロックを持つマスター・デバイス。これは、他のすべてのマスター・デバイス (境界クロックとも呼ばれる) を同期化し、次に、マスター・デバイスが関連付けられたすべてのスレーブ・デバイスを更新して同期化します。グランドマスター・クロックは、ベスト・マスター・クロックまたはベスト・クロックとも呼ばれます。

フラグ

ショート・フラグ	ロング・フラグ	説明
-a	--delay-override	スレーブ状態のマスターによって告知された遅延要求間隔をオーバーライドします。このフラグは、 ptpengine 属性を次の設定値に設定します。 ptpengine:log_delayreq_override
-A	--auto-lock	事前設定されたまたはポート・モード固有のロック・ファイル名を使用します。このフラグは、 ptpd デーモンの複数インスタンスを実行する場合に便利です。
-c file	--config-file file	構成ファイルのパスを指定します。
-C	--foreground	フォアグラウンドで実行するコマンドを指定します。このフラグは、 global 属性を次のように設定します。 global:foreground=Y
-d domain_number	--domain domain_number	注: ptpd デーモンが SRC から開始されている場合、このオプションは無視されます。 所属する PTP ドメイン番号を指定します。このフラグは、 ptpengine 属性を次のように設定します。 ptpengine:domain
-D [DD]	--debug	デバッグ・レベルを指定します。このフラグは、 global 属性を次のように設定します。 global:debug_level デバッグ・レベル情報を増やすには、複数インスタンスを指定できます。例えば、 -D オプションは、最小デバッグ情報を指定します。 -DD オプションは、詳細デバッグ情報を指定します。 -DDD オプションは、最大デバッグ情報を指定します。
-e setting	--explain setting	単一の設定のヘルプ情報を表示します。このフラグは、 section 属性を次のように設定します。 section:key
-E	--e2e	エンドツーエンド遅延検出を指定します。このフラグは、 ptpengine 属性を次のように設定します。 ptpengine:delay_mechanism=E2E
-f file	--log-file file	ログ・ファイルへのパスを指定します。このフラグは、 global 属性を次のように設定します。 global:logfile

ショート・フラグ	ロング・フラグ	説明
-h	--help	ヘルプ画面を表示します。
-H	--long-help	すべての設定および動作の詳細ヘルプを表示します。
-i dev	--interface dev	ptpd 実装に使用するインターフェースを指定します。例えば、en0 になります。このフラグは、ptpengine 属性を次のように設定します。 ptpengine:interface PTP 構成と出口を検査します。構成が正しい場合は 0 を返します。
-k	--check-config	ロック・ファイルのパスを指定します。このフラグは、global 属性を次のように設定します。
-l file	--lockfile file	global:lock_file ロック・ファイルの検査とロックングをスキップします。このフラグは、global 属性を次のように設定します。
-L	--ignore-lock	global:ignore_lock 完全な IEEE 1588 実装 (グランドマスター (ベスト・マスター) でない場合、マスター、スレーブ) を指定します。このフラグは、ptpengine 属性を次のように設定します。
-m	--masterslave	ptpengine:preset=masterslave マスターのみモード (ベスト GM でない場合、パッシブ) を指定します。このフラグは、ptpengine 属性を次のように設定します。
-M	--masteronly	ptpengine:preset=masteronly クロックを調整しないことを指定します。このフラグは、clock 属性を次のように設定します。
-n	--clock:no_adjust	clock:no_adjust デフォルト構成と出口を表示します。この出力は、構成ファイルとして使用できません。
-O	--default-config	
-p	--print-lockfile	ロック・ファイルのパスと出口を出力します。このフラグは、init スクリプトを自動ロック・ファイルと組み合わせるのに便利です。
-P	--p2p	ピアツーピア遅延検出を指定します。このフラグは、ptpengine 属性を次のように設定します。
-r number	--delay-interval number	ptpengine:delay_mechanism=P2P 遅延要求メッセージ (ログ 2) の間隔を指定します。このフラグは、ptpengine 属性を次のように設定します。
-R dir	--lock-directory dir	ptpengine:log_delayreq_interval ロック・ファイルを保管するディレクトリーを指定します。このフラグは、global 属性を次のように設定します。
-S file	--statistics-file file	global:lock_directory 統計ファイルのパスを指定します。このフラグは、global 属性を次のように設定します。
-s	--slaveonly	global:statistics_file スレーブのみモードをオンにします。このフラグは、ptpengine 属性を次のように設定します。
-u	--unicast	ptpengine:preset=slaveonly ユニキャスト・モード (ユニキャスト・ネゴシエーションなし) を指定し、すべてのメッセージを IP に送信します。このフラグは、ptpengine 属性を次のように設定します。
-v	--version	ptpengine:ip_mode=unicast + ptpengine:unicast_address バージョン・ストリングと出口を出力します。
-V	--verbose	フォアグラウンドで実行して、すべてのメッセージを標準出力に記録するコマンドを指定します。このフラグは、global 属性を次のように設定します。 global:verbose_foreground=Y

注: **ptpd** デーモンが SRC から開始されている場合、このオプションは無視されます。

ショート・フラグ	ロング・フラグ	説明
-y	--hybrid	ハイブリッド・モード (マルチキャストとユニキャストの操作の混合) を指定します。同期と告知にマルチキャストを使用し、遅延要求と応答にユニキャストを使用します。このフラグは、ptpengine 属性を次のように設定します。
		<code>ptpengine:ip_mode=hybrid</code>

PTP デーモンのポート状態

ptpd ポートには、以下の状態があります。

状態	説明
init	初期化中
flt	障害
lstn_init	listen 中 (初回)
lstn_reset	listen 中 (後続のリセット)
pass	パッシブ (ベスト・マスター以外、告知以外)
uncl	未調整
slv	スレーブ
pmst	事前マスター
mst	マスター (アクティブ)
dsbl	使用不可
? (unk)	不明な状態

統計ログ・ファイルのフォーマット

ptpd 統計ログを使用可能にすると、以下のオプションが選択可能です。

ptpengine:log_statistics

各受信 PTP パケットのログイン情報を更新します。

ptpengine:statistics_file

統計ログ・ファイルのロケーション・パスを指定します。

注: このオプションは、統計の収集を使用可能にします。

統計ロギングを使用可能にすると、**ptpd** スレーブは、同期および遅延応答メッセージを受信したときにクロック同期情報をログに記録します。**ptpd** デーモンがログを開始またはフラッシュすると、コメント行 (# で始まる) がログに記録され、すべての列の名前が入ります。ログ・ファイルは **comma-separated values (CSV)** フォーマットで、統計ツールや表計算ソフトウェア・パッケージに簡単にインポートして、分析やグラフの作成が行えます。**ptpd** デーモンを長期間、高いメッセージ送信率で実行すると、ログ・ファイルのサイズが増えます。ログに記録されるメッセージの数を減らすには、`global:statistics_log_interval` 設定を使用して、ログ出力を構成済み間隔当たり 1 メッセージに制限できます。統計ログ・ファイルのサイズと最大数も制御できます。

統計ログ・ファイル内の列の説明は、次のとおりです。

Timestamp

メッセージを受信した時刻。日時の情報は、`global:statistics_timestamp_format` 設定に応じて、テキスト、UNIX タイム・スタンプ、または両方の形式 (この場合は、フィールドが追加されます) として表されます。ログ・ファイルをプロット作成ソフトウェアにインポートするときに、そのソフトウェアが UNIX 時刻を認識できる場合は、タイム・スタンプの形式を `unix` または `both` に設定してください。一部のソフトウェアは、テキストから日時を変換する際に、秒の小数部分を解釈しないからです。

State ポートの状態。各種のポート状態について詳しくは、558 ページの『PTP デーモンのポート状態』を参照してください。

Clock ID

IEEE 1588 標準で定義された、現行のベスト・マスターのポート ID。ローカル・クロックがベスト・マスターである場合、この ID は、ローカル・クロックの ID です。このパラメーターは、clock_id または port (ホスト) として表示されます。ポートは、PTP クロック・ポート番号であり、User Datagram Protocol (UDP) ポート番号ではありません。クロック ID は、MAC アドレスの中央に 0xfffe を挿入して 48 ビット MAC アドレスから変換された、Extended Unique Identifier (EUI)-64 64 ビット ID です。

One-way delay

スレーブ状態にある **ptpd** デーモンが遅延要求と遅延応答のメッセージ交換から計算した、片方向遅延 (または、平均パス遅延) の現行値 (秒単位) です。

注: この値がゼロのままの場合は、遅延応答メッセージが受信されていないことを意味し、ネットワーク問題に原因がある可能性があります。

Offset from master

マスター・デバイスからの現行のオフセット値 (秒単位)。これは、スレーブ状態にある PTP エンジンのメイン出力です。この値は、クロック・サーボ・アルゴリズムでのクロック修正のための入力となります。この値は通常、スレーブ・デバイスのパフォーマンスの見積もり時に測定されます。

Slave to master

遅延要求と遅延応答のメッセージ交換から抽出された中間オフセット値 (秒単位)。この値は、片方向遅延の計算に使用されます。最後の値がフィルターによってリジェクトされた場合、その前の値がログ・ファイルに表示されます。遅延応答メッセージを受信しなかった場合、この値はゼロ (0) です。

Master to slave

同期メッセージから抽出された中間オフセット値 (秒単位)。この値は、マスター・デバイスからのオフセット値の計算に使用されます。最後の値がフィルターによってリジェクトされた場合、その前の値がログ・ファイルに表示されます。

Observed drift

クロック制御比例積分 (PI) サーボ・モデルの統合アキュムレーターによって測定された、スレーブ・クロックとマスター・クロックの間の周波数の差。クロック・オフセット値が安定すると、この値が安定するので、この値はクロックの安定度を検出するために使用されます。

Last packet received

このフィールドは、最後に受信したメッセージを表示します。同期メッセージの場合は S を表示し、遅延応答メッセージの場合は D を表示します。スレーブ・デバイスのログに D 項目がない場合は、ネットワーク問題のためにスレーブ・デバイスが遅延応答メッセージを受け取っていないことを意味しています。

One-way delay mean

最後のサンプリング・ウィンドウについて計算された片方向遅延の平均。

One-way delay std dev

最後のサンプリング・ウィンドウについて計算された片方向遅延の標準偏差。

Offset from master mean

最後のサンプリング・ウィンドウについて計算されたマスター平均からのオフセット。

Offset from master std dev

最後のサンプリング・ウィンドウについて計算されたマスター標準偏差からのオフセット。

Observed drift mean

最後のサンプリング・ウィンドウについて計算された観測ドリフトまたはローカル・クロック周波数調整の平均

Observed drift std dev

最後のサンプリング・ウィンドウについて計算された観測ドリフトまたはローカル・クロック周波数調整の標準偏差。低い値は、クロックがあまり積極的に制御されていないことを示します。したがって、値がより安定しています。

注: すべての統計測定 (平均および標準偏差) は、**--enable-statistics** フラグを使用して **ptpd** デーモンが作成されていた場合にのみ、計算されて表示されます。サンプリング期間の所要時間は、`global:statistics_update_interval` 設定を使用して制御されます。

シグナルの処理

ptpd デーモンは、以下のシグナルを処理します。

項目	説明
SIGHUP	構成ファイルを再ロードし (デーモンによって使用される場合)、ログ・ファイルを再オープンします。SRC の refresh サブコマンドは同じタスクを実行します。
SIGUSR1	サブシステムがスレープ状態にある場合、 ptpd デーモンは強制的にクロックをマスター値から現行オフセット値にステップさせます。
SIGUSR2	すべての PTP プロトコル・カウンターを現行ログ・ターゲットにダンプします (そして、 <code>ptpengine:sigusr2_clears_counters</code> 属性が設定されている場合は、カウンターをクリアします)。
SIGINT SIGTERM	ログ・ファイルとその他のオープン・ファイルを閉じます。ロック・ファイルもクリーンアップして、終了します。
SIGKILL	非クリーン終了を強制します。

終了状況

終了すると、**ptpd** デーモンは正常時には 0 を返します。これは、デーモン・モードで正常に開始したか、クリーンに終了したかのいずれかです。**-k (--check-config)** オプションが使用されており、構成が正しかった場合も、値 0 が返されます。エラー時には、ゼロ以外の終了コードが返されます。**ptpd** デーモンが root 以外のユーザーによって開始された場合は、値 127 が返されます。ロック・ファイル・エラー時、および **ptpd** デーモンをデーモンとして開始できない場合は、値 3 が返されます。デーモンの開始時のメモリー割り振りエラーでは、値 2 が返されます。構成エラー、ヘルプ・モードまたはパラメーターなしでの **ptpd** デーモンの実行、サブシステムの自動シャットダウン、ネットワーク始動エラーなど、その他のすべてのエラー条件では、値 1 が返されます。

例

- SRC で **ptpd** デーモンを開始するには、次のコマンドを入力します。
`startsrc -s ptpd`
- SRC で **ptpd** デーモンを停止するには、次のコマンドを入力します。
`stopsrc -s ptpd`
- SRC で **ptpd** デーモンをリフレッシュするには、次のコマンドを入力します。
`refresh -s ptpd`

ptpd デーモンは構成ファイルを再ロードし (デーモンによって使用される場合)、ログ・ファイルを再オープンします。

4. /etc/ptpd2.conf パス内の構成ファイルが正しく構成されているかどうかを確認するには、次のコマンドを入力します。

```
ptpd -k
```

5. 単一の設定の意味を表示するには、次のコマンドを入力します。

```
ptpd -e ptpengine:interface
```

この出力は、ptpengine:interface 設定の意味を説明します。

ファイル

項目	説明
/etc/ptpd2.conf	ptpd デーモン構成ファイルのデフォルト・パス。
/usr/samples/tcpip/ptpd2/ptpd2.conf	ptpd2.conf 構成ファイルのサンプル・ファイル。

ptsc コマンド

目的

openpts ベリファイヤーからの認証要求に備えて、トラステッド・プラットフォーム・モジュール (TPM) から情報を収集します。

構文

```
ptsc [ options ] [ commands ]
```

説明

ptsc コマンドは openpts コレクターです。このコマンドは、TPM から (tscd インターフェースを介して) 測定値とイベントを収集し、リファレンス・マニフェスト (RM) を作成し、要求された場合にそれらを openpts ベリファイヤーに伝えるために使用されます。システムが初めてトラステッド・ブート用に構成される際には、**-i** オプションを使用してコレクターを初期化する必要があります。このオプションは、/var/ptsc/<UUID>/rm0.xml ファイルに保管された UUID および関連 RM を生成します。システムが変更されて新しい RM が必要な場合には、**-u** オプションを使用してベリファイヤーを再初期化する必要があります。

フラグ

項目	説明
コマンド	
-i	openpts コレクターを初期化します。
-s	起動を指定します (自己診断テストおよびタイム・スタンプの両方)。
-t	自己診断テストを示します。
-u	RM を更新します。
-U	RM を自動的に更新します。
-D	ターゲットまたはすべてのオプションの構成設定を表示します。これがデフォルト設定です。
-m	-M モードの場合
オプション	
-c configfile	構成ファイルのロケーションを変更します。デフォルトは /etc/ptsc.conf です。
-P name=value	プロパティを設定します。
-R	RM を除去します。
-Z	すべてゼロの SRK 秘密情報を使用します。
その他	

項目	説明
-h	コマンド使用方法の情報を表示します。
-V	情報を詳細モードで表示します。複数の -V オプションは詳細度を高め、デバッグに使用されます。

ファイル

項目	説明
/etc/ptsc.conf	構成ファイル。これは構成ファイルのデフォルト・ロケーションです。
/var/ptsc/rm-uuid	現行 RM の UUID。
/var/ptsc/uuid	コレクターの UUID。
/var/ptsc/<UUID>/rm0.xml	リファレンス・マニフェスト。

ptsevt コマンド

目的

AIX システム・ブート・イメージに対する更新の通知を管理します。

構文

```
ptsevt [ -a ] [ -r ] [ host port]
```

```
ptsevt -c
```

```
ptsevt [ -u uuid ] -e
```

説明

ptsevt ユーティリティーは、**-e** オプションを使用することによって、リスナーとして知られる認証ソフトウェアがサブスクライブできるブート・イメージ更新に関するイベントを送信します。オプションの **-u** 引数を使用すると、更新されている AIX システムのコレクターの汎用固有 ID (UUID) を指定することができます。**-u** 引数を指定しないと、**ptsevt** コマンドは **/var/ptsc/uuid** ファイル内で見つかったデフォルト値を使用します。

サブスクライバーは、**-a** および **-r** オプションを使用して、それぞれ追加または除去することができます。ホストにはシンボリック・アドレスか、IP または IPv6 番号を使用でき、TCP ポートは 10 進数でなければなりません。

-c オプションは、サブスクリプション・リストを消去するために使用します。

フラグ

項目	説明
-a	ホストおよびポート引数によって指定されたリスナーを、サブスクライバー・リストに記述された宛先に追加します。
-c	サブスクライバーのリストを消去します。
-e	イベント通知をリスト内のすべてのサブスクライバーに送信します。
-r	ホストおよびポート引数によって指定されたリスナーを、サブスクライバーのリストから除去します。
-u	通知の一部として送信される UUID を指定します。デフォルトでは、 ptsevt コマンドは /var/ptsc/uuid ファイル内で見つかった値を使用します。

ファイル

項目	説明
/var/ptsc/subscribers	サブスクライバー・リスト。
/var/ptsc/subscribers.lo ck	サブスクライバー・リスト・ロック・ファイル。
/var/ptsc/uuid	通知の一部として送信されるデフォルトの UUID。

ptsevtd コマンド

目的

AIX システム・ブート・イメージに対する更新の通知を管理します。

構文

```
ptsevtd [ -c command ] [ -d ] [ -f ] [ -p port name]
```

説明

認証されたシステムが更新されている場合、**ptsevtd** デーモンは **ptsevt** コマンドによって送信されたイベントを **listen** します。デフォルトでは、イベントが受信されると必ず、**ptsevtd** コマンドはそのイベントを送信しているシステムの汎用固有 ID (UUID) を最初の引数として **openpts** コマンドを呼び出します。このプロセスは対応するリファレンス・マニフェストを、最新または期待される測定値で更新します。-c オプションを使用すると、通知の受信時に呼び出される代替コマンドを指定することができます。

-f オプションは、デーモンをフォアグラウンドで実行するために使用します。-d オプションを複数回指定すると、出力がより詳細になります。-p 引数は、イベント通知の **listen** に使用するポートを指定します。

フラグ

項目	説明
-c	通知を受信した場合に呼び出すコマンドを指定します。このオプションを指定しないと、デフォルトで openpts コマンドが使用されます。
-d	出力の詳細度を高めるレベルを指定します。
-f	リスナーをフォアグラウンドで実行します。出力は stderr コンソールに送信されます。
-p	イベント通知に使用する TCP ポートを指定します。デフォルトは 34185 です。

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

項目	説明
-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。
-f	ソートのときに大文字と小文字を区別しません。
-g <i>Number</i>	出力されるときに ptx 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。
-i <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。 -i と -o フラグを使用しない場合には、 <i>/usr/lib/eign</i> ファイルがデフォルトの <i>Ignore</i> ファイルです。
-o <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。
-r	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。
-t	フォントタイプセッター用の出力を準備します。
-w <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、 nroff コマンドの場合には 72 文字で、 troff コマンドの場合には 100 文字です。

項目	説明
—	(二重ダッシュ) フラグの終わりを示します。

パラメーター

項目	説明
<i>Infile</i>	英語テキストを指定します。標準入力デフォルトのファイルです。 ptx 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>	ptx 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 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。

ファイル

項目	説明
<i>/usr/lib/eign</i>	デフォルトの <i>Ignore</i> ファイルが入っています。
<i>/usr/share/lib/tmac/tmac.ptx</i>	マクロ・ファイルが入っています。

関連資料:

230 ページの『**nroff** コマンド』

関連情報:

troff コマンド、**mm** コマンド、**mptx** コマンド

| **pvcauth** コマンド

| 目的

| **pvcauth** コマンドは、IBM Power® Virtualization Center (PowerVC) で認証を行ってトークンを取得するために使用されます。このトークンは AIX Live Update操作に PowerVC サービスを使用するために必要となります。このトークンは設定された期間でのみ有効です。また、このコマンドはトークンの無効化にも使用できます。

| 構文

| PowerVC で認証を行ってトークンを取得するには、次の構文を使用します。

```
| pvcauth [ -u user_name ] [ -p password ] -a pvc [ -o project ][ -P port ]
```

| 既に収集されているトークンを無効化して削除するには、次の構文を使用します。

```
| pvcauth -r -a pvc
```

| 既知の PowerVC 認証トークンをすべてリストするには、次の構文を使用します。

```
| pvcauth -l
```

| 説明

| **pvcauth** コマンドを使用できるのは、すべてのタイプのオブジェクトにアクセスできる場合、および適切な PowerVC 管理権限がある場合です。 **pvcauth** コマンドは、AIX パーティション管理者が Live Update操作の実行に使用できるトークンを生成します。このコマンドが正常に終了すると、トークンがカーネルに保管されます。この時点で、**geninstall** コマンドを使用して Live Update操作を実行できるようになります。

| このコマンドを使用するには、以下のタスクを実行する権限が必要です。

- | • 管理対象パーティションの電源をオンにする。
- | • 管理対象パーティションをシャットダウンする。
- | • 管理対象パーティションを作成する。
- | • 管理対象パーティションを削除する。
- | • ストレージ・ボリュームを管理する。
- | • ネットワーク・アダプターを管理する。

| パラメーター

| **password**

| パスワードを指定する最大 64 文字のストリング。

| **port**

| PowerVC に接続するポート番号を指定する最大 16 文字のストリング。このパラメーターのデフォルト値は 5000 です。

| **project**

| PowerVC プロジェクト名を指定する最大 64 文字のストリング。

| **pvc**

| 認証に PowerVC のホスト名または IP アドレスを指定する最大 64 文字のストリング。

| **user_name**

| PowerVC ユーザー名を指定する最大 64 文字のストリング。

| フラグ

| **-a pvc**

| これは認証用に PowerVC のホスト名または IP アドレスを指定します。

| **-o project**

| これは、PowerVC で認証を行うために使用される PowerVC プロジェクト名を指定します。 **-o** フラグを指定しない場合、プロジェクト名はデフォルト名 **ibm-default** に設定されます。

| **-l** これは既知の PowerVC 認証トークンをすべてリストします。リストされる情報には、トークンの現行存続時間 (TTL) 値が含まれます。

| **-p password**

| これは認証用の PowerVC パスワードを指定します。 **-p** フラグを指定しない場合は、**pvcauth** コマンドの実行時にパスワードが要求されます。

| **-P port**

| これは、PowerVC に接続するために使用できるポート番号を指定します。

| **-r** これは PowerVC で生成されたトークンを削除します。

| **-u user_name**

| これは認証に使用できる PowerVC ユーザー名を指定します。すべてのタイプのオブジェクトに対するアクセス権限と、適切な PowerVC 管理権限が必要となります。

| 例

| 1. **apollo** という HMC で認証を行うとします。この HMC にはファイアウォールがあります。また、この HMC では、PowerVC ポート 5000 はアクセス不可です。この場合、再バインド・プロキシ

| ー・ノードをセットアップすれば、開いている別のポートを使用できます。論理パーティション
| mylpar で認証を行って、プロキシ・ノード proxy1 上のポート 14111 で SSH クライアントを使用
| するには、以下のコマンドを入力します。

```
| root @ proxy1: /  
| # ssh -R localhost:14111:apollo:5000 root@mylpar  
|  
| root @ mylpar: /  
| # pvcauth -a localhost -u hscroot -P 14111  
| Enter HMC password:
```

| **lvupdate.data** ファイルの pvc スタンザで management_console 属性を localhost として指定すれ
| ば、ライブ・アップデート操作を開始できます。

| 2. IP アドレス 5.5.55.121 の PowerVC でパスワード・プロンプトにより認証を行うには、次のコマンド
| を入力します。

```
| # pvcauth -a 5.5.55.121 -u root  
| Enter password for root:
```

| 3. IP アドレス 5.5.55.121 の PowerVC で行われた以前の認証を無効化するには、次のコマンドを入力し
| ます。

```
| # pvcauth -r -a 5.5.55.121
```

| 関連情報:

| PowerVC 管理オブジェクトを NIM 環境に追加

pvi コマンド

目的

特権ファイルにアクセスできるように、特権エディターを提供します。

構文

```
pvi [ -l ] [ -R ] [ -w Number ] [ -c | + [ Subcommand ] ] [ File ]
```

説明

pvi コマンドは、*File* パラメーターで指定されたファイルを編集するために、**vi** エディターの特権バージョンである **pvi** エディターを呼び出します。一度に 1 つのファイルのみを開くことができます。そのファイルは、特権ファイル・データベースで定義されたセキュリティー属性を持っていなければなりません。このエディターでは、権限の 1 つ以上がファイルの **readauths** 属性または **writeauths** 属性内の権限の 1 つ以上に一致する場合にのみ、そのファイルを表示することができます。その場合にバッファの内容の変更が可能になります。このエディターでは、権限の 1 つ以上がファイルの **writeauths** 属性内の権限の 1 つ以上に一致する場合にのみ、そのファイルに書き込むことができます。 **pvi** コマンドで開いたファイルは、開いたときと同じパスにのみ書き込むことができます。

pvi エディターの開始と終了はコマンド・モードで行いますが、テキストを追加または変更するには、テキスト入力モードに切り替える必要があります。テキスト入力モードを開始するサブコマンドについては、テキスト入力モードの項を参照してください。テキストをファイルに保存するには、**:w** コマンドの 1 つを使用します。 **pvi** エディターを終了するには、**:q** コマンドを使用します。

この **pvi** コマンドで開始されるフルスクリーン表示エディターは、**ex** エディターを基にしています。 **pvi** エディターの中で **ex** サブコマンドを実行できます。それらのサブコマンドは表示画面のカーソル位置で機能します。

pvi エディターは、編集中のファイルのコピーを編集バッファに作成します。変更を保存するまでは、ファイルの内容は変更されません。

注: **vi** エディターの機能の一部は、**pvi** エディターでは使用できません。 **vi** エディター上の情報を参照する場合、**-r** フラグ、**-t** フラグ、シェル・エスケープ、ユーザー定義マクロ、キー・マッピング、および **vi** オプションの永続設定は **pvi** エディターではサポートされないことを知っている必要があります。一度に開くことができるバッファは 1 つのみです。また、ファイルは開いたときと同じパスにのみ書き込むことができます。

エディターの制限

pvi エディターで扱える文字数には次の上限があります (1 バイト文字の数)。

- グローバル・コマンド・リスト当たり 256 文字
- シェル・エスケープ・コマンドごとに 2048 文字
- 単一値オプション当たり 128 文字
- タグ名当たり 30 文字
- 非表示強制 524,230 行
- マップ・マクロ 128、合計 2048 文字

編集モード

pvi エディターの操作モードは次のとおりです。

項目	説明
コマンド・モード	pvi エディターはコマンド・モードで開始されます。テキスト入力モードでテキストを訂正するサブコマンドを除くすべてのサブコマンドを呼び出すことができます。サブコマンドの説明については、「 pvi エディターのサブコマンド」の各トピックを参照してください。コマンド・モードから呼び出せないサブコマンドを識別するには、入力モードでのテキストの変更を参照してください。サブコマンドおよびその他のモードが終了すると、 pvi エディターはコマンド・モードに戻ります。1 つのサブコマンドを取り消すには、 Esc キーを押します。
テキスト入力モード	テキストの追加または変更のために許可されているコマンドを使用すると、 pvi エディターはテキスト入力モードに入ります。テキスト入力モードを開始するサブコマンドのリストを見るには、ファイルへのテキストの追加を参照してください。また、コマンド・モードでテキストを変更するサブコマンド、 C サブコマンド、および cx サブコマンドも参照してください。これらのサブコマンドの 1 つを入力した後、テキスト入力モードで機能するサブコマンドのいずれかを使用してテキストを編集することができます。サブコマンドのリストを見るには、 pvi エディターのサブコマンドの各トピックを参照してください。テキスト入力モードからコマンド・モードに戻るには、 Esc キーを押すか (通常の終了の場合)、または Ctrl + C キーを押します (INTERRUPT シグナルを作成する場合)。
最終行モード	サブコマンドには、画面の最下行に表示された入力を読み取るものもあります。それらは、接頭部としてコロン (:)、スラッシュ (/)、または疑問符 (?) を持つサブコマンドです。接頭部文字を入力すると、 pvi エディターによりカーソルが画面の最下部に移動し、コマンドの残りの文字を入力できるようになります。サブコマンドを実行するには、 Enter を押します。サブコマンドを取り消すには、 Ctrl + C を押して、 INTERRUPT シグナルを作成します。コロン (:) を使用して最終行モードに入った場合、以下の文字は、カウントを指定するコマンドで使用すると特殊な意味を持ちます。 % カーソル位置に関わりなくすべての行 \$ 最終行 . 現在行

pvi エディターのカスタマイズ

pvi エディターは、**vi** エディター・オプションの設定の指示に従って、一時的にカスタマイズできます。

pvi エディターのサブコマンド

pvi エディターに適用できる **vi** エディター・サブコマンドについては、以下のリストに情報がありません。

- **vi** 一般サブコマンドの構文
- 画面調整用の **vi** サブコマンド。
- **vi** エディターによるテキスト編集。
- **vi** エディターによるファイル操作。
- **vi** エディターを中断および終了するためのサブコマンド。

フラグ

項目	説明
-c [<i>Subcommand</i>]	編集が開始される前に ex エディター・サブコマンドを実行します。これにより行指向のテキスト・エディターが提供されます。 <i>Subcommand</i> パラメーターにヌル・オペランドを指定する (例えば -c '' と指定する) と、エディターはファイルの最終行にカーソルを置きます。
-l	エディターをリスト処理 (LISP) モードにします。このモードでは、エディターは LISP モードに適するようにインデントを行い、また (、)、{ 、[[、および]] サブコマンドは LISP で動作するように変更されます。これらのサブコマンドは指定された LISP 関数にカーソルを位置付けます。LISP サブコマンドについて詳しくは、文、段落、およびセクションへの移動を参照してください。
-R	ファイルの上書き防止のために readonly オプションを設定します。
-w <i>Number</i>	デフォルト・ウィンドウ・サイズを <i>Number</i> パラメーターで指定された値に設定します。これは低速回線を介してエディターを使用するときに役立ちます。
+[Subcommand]	-c サブコマンドと同じです。

セキュリティ

アクセス制御: このコマンドは、すべてのユーザーに実行 (x) アクセス権限を与えます。

役割ベースのアクセス制御: ユーザーが特権ファイル・データベースの **readauths** または **writeauths** 権限リストにある権限に一致する権限を持っている場合、このコマンドはファイルへの読み取りアクセスを認可します。ユーザーが特権ファイル・データベースの **writeauths** 権限リストにある権限に一致する権限を持っている場合のみ、このコマンドはファイルへの書き込みアクセスを認可します。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

1. 特権エディターを呼び出して **plans** ファイルを編集するには、次のように入力します。

```
pvi plans
```

このコマンドは **pvi** エディターをコマンド・モードに置きます。テキストを追加または変更するには、テキスト入力モードに入るか、あるいはコマンド・モードで受け入れられるコマンドを使用する必要があります。

2. **pvi** エディターで作成したテキストを保存するには、**Esc** を押してテキスト入力モードを終了してから、次のように入力します。

:w

3. テキスト入力モードの **pvi** エディターを終了するには、**Esc** を押してコマンド・モードに入ってから、次のように入力します。

:q!

エディターが既にコマンド・モードになっている場合は、終了 (**q!**) コマンドを出す前に **Esc** を押す必要はありません。

ファイル

項目	説明
<code>/usr/bin/pvi</code>	pvi コマンドが入っています。
<code>/etc/security/privfiles</code>	特権ファイルのセキュリティー属性が入っています。

関連情報:

ex コマンド

vi コマンド

ロールベースのアクセス制御 (RBAC)

ネットワークの保護

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ます。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

項目		説明
-d <i>DebugLevel</i>		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 <i>Protocol</i>		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。
		HMAC-MD5
		HMAC-MD5 認証プロトコルで使用する keychange 値を生成する。
		HMAC-SHA
		HMAC-SHA 認証プロトコルで使用する keychange 値を生成する。
		all HMAC-MD5 および HMAC-SHA の両方の keychange 値を生成する。
		デフォルトでは、HMAC-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・ラインにカット・アンド・ペースト操作できるようにするためです。
-u <i>KeyUsage</i>		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

項目		説明
<i>EngineID</i>		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。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>		鍵を使用する宛先ホストを指定します。
<i>IPAddress</i>		鍵を使用する宛先ホストでエージェントの IPv4 または IPv6 アドレスを指定します。
<i>NewPassword</i>		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。
<i>OldPassword</i>		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。

セキュリティー

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティー」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

pwchange コマンドは、選択したプロトコルと鍵の用途により、異なる出力を生成します。通常 **Keychange** 値は、変更される鍵の 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円記号 (¥) 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、鍵および **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。

2. 以下のコマンドは、IPv6 アドレス指定の **pwchange** の使用法を示しています。

```
pwchange oldpassword newpassword 2000:1:1:1:209:6bff:feae:6d67
```

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:  
0000774adc53ba4b0427dc2f65568435  
721847d1b5cb597daa85d003033afba3
```

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.21.128.0.0.2.2.32.0.0.1.0.1.0.1.2.9.107.255.254.174.  
109.103.6.105.112.118.54.117.49 ¥'36133c694155026620637761f835ef616de294f37f758c74ff1544ca3de279b8¥'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円記号 (¥) 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。このケースでは、EngineID は 21 オクテット 128.0.0.2.2.32.0.0.1.0.1.0.1.2.9.107.255.254.174.109.103 です。また、ユーザー名の ASCII 表記は、このケースでは 6 文字の長さであり、105.112.118.54.117.49 に変換されます。

注: **pwchange** コマンドは、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。

関連資料:

581 ページの『pwtokey コマンド』

関連情報:

clsnmp コマンド

snmpdv3 コマンド

/etc/clsnmp.conf コマンド

/etc/snmpdv3.conf コマンド

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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

1. 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

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。

項目	説明
0	正常終了。
>0	エラーが発生しました。

例

次のように入力すると、

```
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

項目	説明
/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 コマンド

ディレクトリー・コマンド

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ー・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。これにより、ユーザーは次に **su** コマンドが与えられると、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になります。

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、セキュリティー・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ー・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ー・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

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ー・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

項目	説明
-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 <i>Flags</i>	パスワードの flags 属性を指定します。 <i>Flags</i> 変数は、以下のリストからの属性をコンマ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 <code>/etc/security/user</code> 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 <code>root</code> 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 <code>root</code> 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。 <i>User</i> 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 <i>load_module</i>	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。

セキュリティー

アクセス制御: `root` ユーザーとセキュリティー・グループのメンバーのみが、このコマンドへの実行 (x) アクセス権限を持ちます。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	<code>/etc/passwd</code>
rw	<code>/etc/security/passwd</code>
r	<code>/etc/security/user</code>

監査イベントは次のとおりです。

イベント	情報
PASSWORD_Change	ユーザー
PASSWORD_Flags	ユーザー、フラグ

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
<code>/usr/bin/pwdadm</code>	pwdadm コマンドが入っています。
<code>/etc/security/passwd</code>	パスワード情報が入っています。
<code>html</code>	

関連資料:

312 ページの『`passwd` コマンド』

関連情報:

ネットワークの保護

`su` コマンド

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 } [-l]{ 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、ユーザー名の間をスペース 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。

フラグを指定して、システムが間違っただけの属性を修正すべきかどうかを指示する必要があります。

/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。

項目	説明
entry	各エントリが読み取り可能で、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、エントリ全体が廃棄されます。
passwd	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 /etc/security/passwd ファイルに転送し、 /etc/security/passwd ファイルで lastupdate 属性を変更して、 /etc/passwd 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、 /etc/security/user ファイルで minalpha 、 minother 、または minlen 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。
ユーザー	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、+ (正符号)、: (コロン)、- (負符号)、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。

項目	説明
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。
lastupdate	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。 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。

項目	説明
auth1	<p>ローカル・ユーザーについて定義された各 <code>SYSTEM;username</code> エントリーが <code>/etc/security/passwd</code> ファイル内に <code>username</code> 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 <code>/etc/security/passwd</code> ファイルに以下のフォーマットでスタanzasが追加されます。</p> <pre>username: password = *</pre> <p>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 <code>/etc/security/user</code> 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。</p> <pre>auth1 = SYSTEM;user</pre> <p>注: auth1 属性は推奨されないので、使用しないでください。</p>
auth2	<p>ローカル・ユーザーについて定義された各 <code>authnameusername</code> エントリーが <code>/etc/security/passwd</code> ファイル内に <code>username</code> 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。</p> <p>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 <code>/etc/security/user</code> 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。</p> <pre>auth2 = NONE</pre> <p>ALL を指定すると、pwdck コマンドは <code>/etc/security/passwd</code> ファイル内の各スタanzasが <code>/etc/security/user</code> ファイル内の <code>SYSTEM;username</code> 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、<code>/etc/security/user</code> ファイル内の <code>username</code> エントリーに対応しないスタanzasが、<code>/etc/security/passwd</code> ファイルから破棄されます。</p> <p>pwdck コマンドは、更新時に <code>/etc/passwd</code> ファイルと <code>/etc/security/passwd</code> 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。</p> <p>pwdck コマンドは、現在の pwdck プロセスの実行中に <code>/etc/passwd</code> ファイルと <code>/etc/security/passwd</code> 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 <code>/etc/passwd</code> ファイルと <code>/etc/security/passwd</code> ファイルを更新します。</p> <p>注: pwdck コマンドは、エラーを修正して報告すると、ファイル上のすべての拡張アクセス制御リスト (ACL) を使用不可にします。</p> <p>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 (<code>/etc/passwd.nm.idx</code>、<code>/etc/passwd.id.idx</code>、<code>/etc/security/passwd.idx</code>、および <code>/etc/security/lastlog.idx</code> 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。<code>/etc/security/lastlog.idx</code> が <code>/etc/security/lastlog</code> 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・ファイルが最新の状態でない場合は、root ユーザーが mkpasswd コマンドを実行するよう指示する警告メッセージが表示されます。</p> <p>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、このコマンドを入力できるのは、root ユーザーあるいはセキュリティー・グループのメンバーです。</p> <p>注: auth2 属性は推奨できなく、使用するべきではありません。</p>

フラグ

項目	説明
-l	実行全体を通して、ファイルをロックします。
-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

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・グループのメンバーに実行 (x) アクセス権限を与えます。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
rw	/etc/passwd
r	/etc/security/user
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

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**Issecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

- 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。

- 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

項目	説明
/usr/bin/pwdck	pwdck コマンドが入っています。
/etc/security/passwd	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。
/etc/security/login.cfg	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
ネットワークの保護

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** プロシーチャーは、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チャーは、 **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

項目	説明
-d <i>DebugLevel</i>	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。

項目
-p Protocol

説明
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。

HMAC-MD5

HMAC-MD5 認証プロトコルで使用する鍵を生成する。

HMAC-SHA

HMAC-SHA 認証プロトコルで使用する鍵を生成する。

all HMAC-MD5 および HMAC-SHA の両方の鍵を生成する。

デフォルトでは、HMAC-MD5 プロトコルの鍵が生成されます。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・ラインにカット・アンド・ペースト操作できるようにするためです。

-s

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

項目
EngineID

説明
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。SNMP 要求で鍵を使用する SNMP エージェントを指定します。SNMP 要求時に鍵を使用する SNMP エージェントの IPv4 または IPv6 アドレスを指定します。

HostName

IPAddress

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。

Password

注: このパスワードは、コミュニティ・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。

セキュリティ

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の

『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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 つの鍵を生成し、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鍵が 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鍵は、**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、認証鍵と秘密鍵、両方に使用する HMAC-SHA 鍵を要求しています。

```
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
```

秘密鍵の出力は認証鍵の出力と同じですが、ローカライズされた秘密鍵は、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。

3. 以下の例では、**pwtokey** コマンドが IPv6 アドレスを使用していることを示しています。

```
pwtokey testpassword 2000:1:1:1:209:6bff:feae:6d67
```

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:  
2a30fe53690fa6b62dba3f9ea30e11fb
```

この例が示すように、**pwtokey** コマンドは 2 つの鍵を生成します。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鍵が 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鍵は、**clsnmp** コマンドの構成で使用されます。SNMP 要求時に鍵を使用する SNMP エージェントは IPv6 アドレスです。

関連資料:

570 ページの『`pwchange` コマンド』

関連情報:

`clsnmp` コマンド

`snmpdv3` コマンド

`/etc/clsnmp.conf` コマンド

`/etc/snmpdv3.conf` コマンド

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

項目	説明
-a	提供される引数。
-f ConfigurationFile	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、 /etc/pxed.cnf 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。

項目	説明
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

項目	説明
/usr/sbin/pxed	PXE Proxy DHCP サーバー・デーモンが含まれています。
/usr/sbin/db_file.dhcpo	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。
/etc/pxed.cnf	pxed デーモンのデフォルトの構成ファイル。

関連情報:

dhcpcsd コマンド

binld コマンド

q

以下の AIX コマンドは、文字 *q* から始まります。

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・システムの起動や停止のほか、ジョブの取り消しも行います。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。

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

項目	説明
-D Printer	<i>Printer</i> 変数で指定したプリンターを起動します。 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 <i>Printer</i> は -D フラグ内の <i>Printer</i> 変数に一致) を入力すると、デバイスの状況は <i>down</i> 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	<i>Printer</i> 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	<i>Printer</i> 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 <i>Printer</i> は -U フラグの <i>Printer</i> 変数と同じ) を入力すると、デバイスの状況は <i>ready</i> 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。

セキュリティー

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティー」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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 1p0 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にプリンター 1p0 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 1p0
```

3. キュー 1p0 に接続されたプリンター 1pd0 を接続するには、次のように入力します。

```
qadm -U 1p0:1pd0
```

ファイル

項目	説明
<code>/usr/sbin/qdaemon</code>	qdaemon 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。

関連資料:

『qcan コマンド』

605 ページの『qprt コマンド』

関連情報:

enq コマンド

`/etc/qconfig` コマンド

印刷キューの開始および停止

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。

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。

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 が 2 回表示される場合がありますが、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 を取り消そうとして、**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、最初に一致したジョブ番号が取り消されますが、そのジョブ番号は、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

項目	説明
-PPrinter	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。
-x JobNumber	<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。
-X	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。 root 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。 root 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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
<code>/usr/sbin/qdaemon</code>	qdaemon 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。

関連資料:

587 ページの『`qadm` コマンド』

605 ページの『`qprt` コマンド』

関連情報:

`enq` コマンド

`/etc/qconfig` コマンド

印刷ジョブの取り消し (`qcan` コマンド)

印刷スプーラー

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、必要な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。

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

項目	説明
<code>-#JobNumber</code>	<i>JobNumber</i> 変数で指定されたジョブ番号の状況を要求します。 <code>-#JobNumber</code> 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 <i>JobNumber</i> を検索します。すべてのキューで <i>JobNumber</i> を検索するには、 <code>-#</code> フラグを <code>-A</code> フラグとともに使用しなければなりません。また、 <code>-#</code> フラグを <code>-P Queue</code> 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、 <code>-P Queue</code> 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。

項目	説明
-A	すべてのキューの状況を要求します。
-L	縦長フォーマットのモードで情報を表示します。 -L フラグと -W フラグが同時に使用される場合、 -L フラグは、印刷ジョブの状況をセミコロンで区切られたフォーマットで表示します。
-P Printer	<i>Printer</i> 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	<i>UserName</i> 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	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フォーマットのモードで情報を表示します。さらに大きなジョブ番号情報がサポートされます。 -W フラグと -L フラグが同時に使用される場合、 -W フラグは、印刷ジョブの状況をセミコロンで区切られたフォーマットで表示します。
-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 <i>Delay</i> 変数によって秒単位で指定された間隔で更新します。

セキュリティ

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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
```

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。

```
qchk -# 123
```

5. 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
```

6. 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

項目	説明
<code>/usr/sbin/qdaemon</code>	qdaemon 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。

関連資料:

587 ページの『`qadm` コマンド』

関連情報:

印刷プーラー

`enq` コマンド

`/etc/qconfig` コマンド

印刷ジョブの状況を検査するコマンド (`qchk` コマンド)

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`、`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。

セキュリティー

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) アクセス権限を持つものとします。

監査イベント
ENQUEUE_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

項目	説明
<code>/usr/sbin/qdaemon</code>	qdaemon 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 qdaemon 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

lpd コマンド

印刷スプーラー

システム・リソース・コントローラー

バックエンドと **qdaemon** の相互作用

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。

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 が 2 回表示される場合がありますが、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

項目	説明
<code>-#JobNumber</code>	保持する印刷ジョブの番号を指定します。
<code>-PQueue</code>	保持する印刷キューを指定します。
<code>-r</code>	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。
<code>-uUser</code>	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。

セキュリティ

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、`lssecattr` コマンドまたは `getcmdattr` サブコマンドの項を参照してください。

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

項目	説明
<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。

関連資料:

605 ページの『`qprt` コマンド』

595 ページの『`qmov` コマンド』

関連情報:

`enq` コマンド

印刷の管理

印刷スプーラー

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。

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 が 2 回表示される場合がありますが、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

項目	説明
-#JobNumber	移動する印刷ジョブのジョブ番号を指定します。
-mNewQueue	宛先の印刷キューの名前を指定します。
-PQueue	移動するジョブの現在の印刷キューを指定します。
-uUser	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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

1. ジョブ番号 280 をキュー lp0 に移動するには、次のように入力します。

```
qmov -mlp0 -#280
```

2. キュー lp1 のすべての印刷ジョブをキュー lp0 に移動するには、次のように入力します。

```
qmov -mlp0 -Plp1
```

3. Mary のすべての印刷ジョブをキュー lp0 に移動するには、次のように入力します。

```
qmov -mlp0 -u mary
```

ファイル

項目	説明
<code>/usr/sbin/qdaemon</code>	qdaemon 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。

関連資料:

593 ページの『`qhld` コマンド』

605 ページの『`qprt` コマンド』

関連情報:

`enq` コマンド

印刷の管理

印刷スプーラー

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。

項目	説明
-s	ServiceCategory 属性の名前 (必須)。
-t	OutgoingTOS 属性。8 ビットの 2 進数で指定。
-b	MaxTokenBucket 属性。Kb (K ビット) で指定。
-f	FlowServiceType 属性 (ControlledLoad または Guaranteed)。
-m	MaxRate 属性。Kbps (秒当たりのキロビット数) で指定。

ポリシー追加のフラグは、以下のとおりです。

項目	説明
-s	ServiceCategory 属性の名前 (必須)。
-r	ServicePolicyRules 属性の名前 (必須)。
-l	PolicyRulePriority 属性 (正整数)。
-n	ProtocolNumber 属性。 <code>/etc/protocols</code> ファイルに定義済み。
-A	SrcAddrRange 属性。a1 から a2 (a2 >= a1) の Source IP アドレス範囲。
-a	DestAddrRange 属性。i1 から i2 (i2 >= i1) の Destination IP アドレス範囲。
-P	SrcPortRange 属性。a1 から a2 (a2 >= a1) の Source Port 範囲。
-p	DestPortRange 属性。i1 から i2 (i2 >= i1) の Destination Port 範囲。

終了状況

項目	説明
0	正常終了。
正整数	エラーが発生しました。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**Issecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

- 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
```
- 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
```
- 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
```
- 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
```

関連資料:

- 601 ページの『qosstat コマンド』
- 599 ページの『qosmod コマンド』
- 600 ページの『qosremove コマンド』
- 598 ページの『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

項目	説明
0	正常終了。
正整数	エラーが発生しました。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**Issecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

- 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
```
- 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
```
- 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
```
- 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
```

関連資料:

- 601 ページの『qosstat コマンド』
- 599 ページの『qosmod コマンド』
- 600 ページの『qosremove 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、ユーザーは見ることができません。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。

項目	説明
-s	ServiceCategory 属性の名前 (必須)。
-t	OutgoingTOS 属性。8 ビットの 2 進数で指定。
-b	MaxTokenBucket 属性。Kb (K ビット) で指定。
-f	FlowServiceType 属性 (ControlledLoad または Guaranteed)。
-m	MaxRate 属性。Kbps (秒当たりの K ビット数) で指定。

ポリシー変更のフラグは、以下のとおりです。

項目	説明
-s	ServiceCategory 属性の名前 (必須)。
-r	ServicePolicyRules 属性の名前 (必須)。
-l	PolicyRulePriority 属性 (正整数)。
-n	ProtocolNumber 属性。 <code>/etc/protocols</code> ファイルに定義済み。
-A	SrcAddrRange 属性。 a1 から a2 (a2 >= a1) の Source IP アドレス範囲。
-a	DestAddrRange 属性。 i1 から i2 (i2 >= i1) の Destination IP アドレス範囲。
-P	SrcPortRange 属性。 a1 から a2 (a2 >= a1) の Source Port 範囲。
-p	DestPortRange 属性。 i1 から i2 (i2 >= i1) の Destination Port 範囲。

終了状況

項目	説明
0	正常終了。
正整数	エラーが発生しました。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

- 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
```
- 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
```
- 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
```
- 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
```

関連資料:

- 601 ページの『qosstat コマンド』
- 598 ページの『qoslist コマンド』
- 『qosremove コマンド』
- 596 ページの『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

項目	説明
0	正常終了。
正整数	エラーが発生しました。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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
```

関連資料:

- 『qosstat コマンド』
- 599 ページの『qosmod コマンド』
- 598 ページの『qoslist コマンド』
- 596 ページの『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

項目	説明
-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:

TCP/IP サービス品質 (QoS)

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。

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。

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 が 2 回表示される場合がありますが、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

項目	説明
-#JobNumber	優先順位を変更するジョブの番号を指定します。
-a PriorityNumber	<i>JobNumber</i> 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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権について詳しくは、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
/usr/sbin/qdaemon	qdaemon デーモンが入っています。
/var/spool/lpd/qdir	ジョブ記述ファイルが入っています。
/var/spool/lpd/stat	デバイスの状況に関する情報が入っています。
/var/spool/qdaemon	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。
/etc/qconfig	構成ファイルが入っています。
/etc/qconfig.bin	/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。

関連資料:

587 ページの『qadm コマンド』

588 ページの『qcan コマンド』

関連情報:

enq コマンド

/etc/qconfig コマンド

印刷ジョブに優先順位を付けるコマンド (qpri コマンド)

qprt コマンド

目的

印刷ジョブを開始します。

構文

```
qprt [ -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 ] [ -I 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

qprt 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。**qprt** 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ければなりません。**-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ければなりません。**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。

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もあります。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。「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 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

項目	説明
-#(j h v)	特殊な機能性を指定します。 <i>Value</i> 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。 <i>OutputBin</i> 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。 <i>PreviewOption</i> 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・ライン・フラグの値であり、コマンド・ラインでフラグ・パラメーターを指定するとオーバーライドされます。 -d に対して指定した (または、デフォルトの) <i>InputDataType</i> 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。

項目

-A Level

説明

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起こ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、**qprt** 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、**qprt** 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にのみ適用されます。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、**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。

項目	説明
-d <i>InputDataType</i>	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。
	<p>a 拡張 ASCII</p> <p>c PCL</p> <p>d Diablo 630</p> <p>g Hewlett-Packard GL</p> <p>p パススルー (変更されずにプリンターに送られます)</p> <p>s PostScript</p>
	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せん。
-D <i>"User"</i>	<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 qprt 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。
-e <i>EmphasizedOpt</i>	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。
-E <i>DblHigh</i>	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。
-f <i>Filter</i>	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、 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。使用可能なフィルター ID は、 pr フィルターを起動する p 、 troff コマンドからの出力を処理する n 、および制御文字を印刷できるようにする l です。
-F <i>Name</i>	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。 <i>Name</i> 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。
-g <i>Begin</i>	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。
-G <i>Coord</i>	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。 <i>Coordinate</i> 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。
-h <i>"Header"</i>	-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。
-H <i>"HostName"</i>	ヘッダー・ページ上のホスト名を設定します。
-i <i>Indent</i>	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、 <i>Indent</i> 変数が含まれなければなりません。

項目	説明
-i <i>FontID</i>	(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。 -iFontID 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。
-iFontPath	(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。
-j <i>Init</i>	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 <i>Restore</i>	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 <i>Color</i>	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 <i>Condense</i>	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 <i>Length</i>	(L の小文字) ページ長を設定します。 <i>Length</i> 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 <i>LineWrap</i>	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。
項目	説明
-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 <i>MessageFile</i>	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 <i>NumberCopies</i>	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 <i>PaperHand</i>	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 <i>Pitch</i>	1 インチ当たりの文字数を設定します。 <i>Pitch</i> 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せん。 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。

項目	説明
-P Queue[:QueueDevice]	<p>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。</p> <ul style="list-style-type: none"> • 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・ライン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	<p>印刷品質を以下のいずれかに設定します。</p> <p>0 高速フォント</p> <p>1 ドラフト品質</p> <p>2 レターに近い品質</p> <p>3 拡張品質</p> <p>300 1 インチ当たり 300 ドット (dpi)</p> <p>600 600 dpi</p>
-Q Value	<p>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、レターサイズ用の紙を表す 1、リーガル・サイズ用の紙を表す 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。</p>
-r	<p>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。</p>
-R Priority	<p>印刷ジョブの優先順位を設定します。Priority 変数の値が大きいくほど、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。</p>
-s NameType	<p>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。</p>
-S Speed	<p>高速印刷を以下のいずれかに設定します。</p> <p>+ 高速印刷を使用します。</p> <p>! 高速印刷を使用しません。</p>
-t TopMargin	<p>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。</p>
-T "Title"	<p>Title 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、qprt コマンド・ライン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。</p>

項目	説明
-u PaperSrc	用紙ソースを以下のいずれかに設定します。 1 基本 2 代替 3 封筒
-U Directional	単一方向印刷を以下のいずれかに設定します。 + 単一方向印刷を使用します。 ! 単一方向印刷を使用しません。
-v LinesPerIn	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 6 と 8 です。
-V Vertical	垂直印刷を以下のいずれかに設定します。 + 垂直印刷を使用します。 ! 垂直印刷を使用しません。
-w PageWidth	ページ幅を文字数単位で設定します。ページ幅には、 -i 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。
-W DbtWide	倍角印刷を以下のいずれかに設定します。 + 倍角印刷を使用します。 ! 倍角印刷を使用しません。
-x LineFeed	自動改行または自動復帰を指定します。 0 改行、垂直タブ、復帰を変更しません。 1 復帰ごとに 1 行だけ改行を追加します。 2 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。
-X CodePage	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。
-y DbtStrike	二重印刷を以下のいずれかに設定します。 + 二重印刷を使用します。 ! 二重印刷を使用しません。
-Y Duplex	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。 0 片面印刷 1 両面印刷、長辺がとじ代 2 両面印刷、短辺がとじ代
-z Rotate	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 (-l) 値と幅 (-w) 値は、これに応じて自動的に調整されます。 0 縦長 1 右向き横長 2 下向き縦長 3 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。

セキュリティー

RBAC ユーザーおよび **Trusted AIX ユーザー**への注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティー」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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・ラインにより、**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・サイズ用の紙に **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

項目	説明
/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。

関連資料:

595 ページの『**qmov** コマンド』

関連情報:

lpr コマンド

/etc/qconfig コマンド

プリンターの初期構成

印刷ジョブを開始するためのコマンド (**qprt** コマンド)

リモート・システムからのファイルの印刷

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・ラインから入力されることなく、**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に戻します。

- **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・ライン・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。

項目

-#JobNumber

説明

JobNumber 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、**-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

/etc/qconfig 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。

項目	説明
-e	qdaemon 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。 -e 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。
-L	長くて詳細なバージョンで状況情報を表示します。 -L フラグと -W フラグが同時に使用される場合、 -L フラグは、印刷ジョブの長い状況をセミコロンで区切られたフォーマットで表示します。
-PPrinter	<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、 LPDEST 環境変数または PRINTER 環境変数のいずれかの値が使用されます。常に LPDEST 変数が PRINTER 変数に優先します。
-q	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、 LPDEST 変数で指定するか、または LPDEST 値がない場合は PRINTER 環境変数で指定します。どちらの変数もない場合には、 qstatus コマンドは、 /etc/qconfig 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。
-u UserName	<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、キューに入られているすべてのジョブの状況が表示されます。
-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サポートされています。 -L フラグと -W フラグが同時に使用される場合、 -W フラグは、印刷ジョブの長い状況をセミコロンで区切られたフォーマットで表示します。
-w DelaySeconds	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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、 **lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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. 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

項目	説明
<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
`rembak` コマンド
`/etc/qconfig` コマンド

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

項目	説明
-i <i>File</i>	<p>標準索引ファイルの代わりに、指定された <i>File</i> を使用します。</p> <p>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。 <i>File</i> 内の行には、以下の構文を使用しなければなりません。</p> <pre>line = category [:category] . . . category = alternate [alternate] . . . alternate = [primary] primary = character [category] option option = {category}</pre> <p>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。</p> <p>¥ (円記号) 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。</p>
-t	<p>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。</p>

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 ¥ (円記号) 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

項目	説明
<code>/usr/games/lib/quiz/index</code>	クイズ・カテゴリ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。

関連資料:

276 ページの『number コマンド』

関連情報:

arithmetic コマンド

back コマンド

ttt コマンド

turnoff 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、JFS ファイルシステムのファイルシステム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 JFS ファイルシステムごとに同じ情報を表示します。

フラグ

項目	説明
-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。 <code>ncheck filesystem sort +0n quot -n filesystem</code>
-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。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**Issecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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
```

- 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
```

- `/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

項目	説明
<code>/etc/passwd</code>	ユーザー名が入っています。
<code>/etc/filesystems</code>	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:

- 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.quotad 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

項目 説明

- 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。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**Issecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
<code>quota.user</code>	ユーザー・クォータを指定します。
<code>quota.group</code>	グループ・クォータを指定します。
<code>/etc/filesystems</code>	ファイルシステムの名前と位置が入っています。

関連資料:

- 『quotacheck コマンド』
- 623 ページの『quotaon または quotaoff コマンド』
- 712 ページの『repquota コマンド』

関連情報:

- クォータ・システム
- edquota 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。

JFS と JFS2 ファイルシステムの両方の場合に、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

項目	説明
-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 と JFS2 の両方)。
-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。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
quota.user	ユーザー・クォータを指定します。
quota.group	グループ・クォータを指定します。
/etc/filesystems	ファイルシステムの名前と位置が入っています。
/etc/group	基本グループ属性が入っています。
/etc/passwd	ユーザー名が入っています。

関連資料:

619 ページの『quota コマンド』

623 ページの『quotaon または quotaoff コマンド』

712 ページの『repquota コマンド』

関連情報:

edquota コマンド

クォータ・システム

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

項目	説明
-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。

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
quota.user	ユーザー・クォータを指定します。
quota.group	グループ・クォータを指定します。
/etc/filesystems	ファイルシステムの名前と位置が入っています。

関連資料:

619 ページの『quota コマンド』

関連情報:

edquota コマンド

クォータ・システム

r

以下の AIX コマンドは、文字 *r* から始まります。

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。

項目	説明
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 人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

項目	説明
?	ヘルプ画面を表示します。
-a <i>Command</i>	実行するアクションを指定します。値は ADD 、 LIST 、 DELETE 、または CHANGE です。
-d <i>Database_filename</i>	データベース・ファイル名を指定します。 radiusd.conf 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。
-e <i>EAP_type</i>	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、 EAP-TLS 、 MD5-challenge 、または none だけを使用できます。デフォルトは、 none です。
-i <i>Config_filename</i>	RADIUS 構成ファイル名を指定します。デフォルトの /etc/radius/radiusd.conf 構成ファイルをオーバーライドするときに使用します。
-l <i>Load_filename</i>	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。
-n	新しいデータベース・ファイルを作成します。 ADD 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。
-p	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・ライン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す。
-t <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、 ADD と CHANGE コマンドを指定する場合に有効です。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。
-u <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。ブランクは含められません。ユーザー ID の重複は認められません。
-w	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。

項目	説明
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・エントリーを表示するには、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 つずつエントリー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、**radiusctl stop** コマンドを使用します。データベースの変更後は、**radiusctl start** 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・ライン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

項目	説明
/usr/radius/bin/raddbm	raddbm コマンドの位置。
/etc/radius/raddbm.bin	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。
/etc/radius/radiusd.conf	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:

セキュア・システムのインストールと構成

radiusctl コマンド

目的

RADIUS 認証、許可、およびアカウントリング用の各デーモンの開始、停止、または再始動を行います。

構文

radiusctl start

radiusctl stop

radiusctl restart

説明

radiusctl コマンドを使用して、RADIUS サーバー・デーモン (ネットワークの認証、許可、およびアカウントリングの制御に使用) の開始、停止、または再始動を行います。

このコマンドを使用すると、AIX Expansion Pack メディア上で出荷される OpenSSL パッケージと一緒に AIX RADIUS サーバーで完全な EAP-TLS のサポートが使用可能となります。

AIX RADIUS サーバーのローカル・ユーザー・データベースは、このサーバーの実行中に更新可能ですが、新規の変更が有効になるのは、そのシステムの再始動後となります。**radiusctl** コマンドを使用するすることも可能です。

注: このコマンドを使用した場合は、AIX RADIUS サーバーの開始と停止に関する古い方法 (例えば、**startsrc -s radiusd**、**stopsrc -s radiusd** など) は、使わないようにします。

フラグ

項目	説明
start	RADIUS サーバーの実行を開始します。 注: OpenSSL を使って EAP-TLS を使用可能にした場合、このサーバーの始動または再始動の試行時に秘密鍵のパスワードを入力するようにプロンプトが出されます。
stop	RADIUS サーバーを停止します。
restart	RADIUS サーバーが現在稼働中であるかどうかとは関係なく、このサーバーを再起動します。このサーバーが稼働中でない場合、このフラグによる動きは start フラグと同じです。

例

1. AIX RADIUS サーバーの稼働を開始するには、以下のコマンドを入力します。
`radiusctl start`
2. 既に実行中の AIX RADIUS サーバーを再始動するには、以下のコマンドを入力します。
`radiusctl restart`
3. 実行中の AIX RADIUS サーバーの稼働を停止するには、以下のコマンドを入力します。
`radiusctl stop`

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

項目	説明
-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。 <i>mode</i> 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)。 <i>mode</i> 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

項目	説明
<code>/usr/ccs/bin/ranlib</code>	ranlib コマンドが入っています。

関連情報:

サブルーチンの概要

`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] [-F]-a

raso -h [*Tunable*]

raso [-F] -L [*Tunable*]

raso [-F] -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 の開発側から具体的な指示がない限り、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

項目	説明
-a	すべてのチューナブル・パラメーターの現在値、リブート値 (-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、パラメーターのリブ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 <i>Tunable</i>	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、Tunable の変更は行われず、警告が表示されます。
-D	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。変更に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。
-F	オプション -a、-L、または -x をコマンド・ラインでのみ指定したとき、制限付きチューナブル・パラメーターの表示を強制します。-F フラグを指定しない場合、表示オプションとの関連で制限付きチューナブルが特に指定されない限り、これらのチューナブルは含まれません。

項目

-h Tunable

-L Tunable

説明

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。

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	4KBpages	D
mtrc_enabled							
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), C (for Connect), and d (for Deprecated)
 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。

-r

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 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。

-x Tunable

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。

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・パラメーターのリポート値です。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。

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に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 のパラメーターを変更すると (-o、-d、または -D フラグを指定)、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という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すると (-o、-d、または -D を指定)、エラー・メッセージが出されます。

タイプ Incremental のパラメーターの現在の値を、現在の値より小さい新しい値に変更すると (-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。

-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。

制限付きチューナブル・パラメーターを変更すると (-o、-d、または -D を指定)、限定使用タイプのチューナブル・パラメーターが変更されたという警告メッセージが表示されます。コマンド・ラインで -r または -p オプションも指定した場合は、変更の確認を求めるプロンプトが出されます。さらに、システム・リブート時に制限付きチューナブル・パラメーターが /etc/tunables/nextboot ファイルにあって、それらの

パラメーターがデフォルト値と異なる値に (コマンド・ラインで **-r** または **-p** オプションを指定して) 既に変更されていた場合は、それらの変更されたチューナブル・パラメーターのリストを示すエラー・ログ・エントリーが作成されます。

省略形 K、M、G、T、P、および E で単位を表して、変更されたチューナブル値を指定することができます。数値の省略形に関連した接頭部および値を次の表に示します。

項目	説明	
省略語	接頭部	2 の累乗
K	キロ	2 ¹⁰
M	メガ	2 ²⁰
G	ギガ	2 ³⁰
T	テラ	2 ⁴⁰
P	ペタ	2 ⁵⁰
E	エクサ	2 ⁶⁰

したがって、1024 のチューナブル値は、1K のように指定されます。

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。

項目	説明
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。パラメーターは Bosboot タイプでなければなりません。

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

sys 0 の **pre520tune** 属性によって制御される 5.2 より前の互換モードで **raso** コマンドを実行する場合、タイプ **Bosboot** のパラメーターを除くパラメーターのリブート値は考慮されません。このモードではブート時にそれらのリブート値は適用されないからです。詳細情報については、「パフォーマンス・マネージメント」の『クライアントでの NFS チューニング』を参照してください。

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ブート値の設定は、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、パラメーターのリブート値は、リブート中に **/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、「*Performance Tools Guide and Reference*」の『カーネル・チューニング』を参照してください。

チューナブル・パラメーター

チューナブルのデフォルト値および値の範囲については、**raso** コマンド・ヘルプ (**-h** <チューナブル・パラメーター名 >) を参照してください。

項目	説明
kern_heap_noexec	<p>目的: 無実行保護をカーネル・ヒープに使用できるようにするかどうかを指定します。</p> <p>チューニング: 保護を使用可能にした場合、保護されたヒープでコーディングを実行しようとするいずれの試みによっても、カーネル例外が生じます。</p>
kernel_noexec	<p>目的: 無実行保護をカーネル・データ領域に使用できるようにするかどうかを指定します。</p> <p>チューニング: 保護を使用可能にした場合、保護された領域でコーディングを実行しようとするいずれの試みによっても、カーネル例外が生じます。</p>
mbuf_heap_noexec	<p>目的: 無実行保護を mbuf ヒープに使用できるようにするかどうかを指定します。</p> <p>チューニング: 保護を使用可能にした場合、保護されたヒープでコーディングを実行しようとするいずれの試みによっても、カーネル例外が生じます。</p>
mtrc_commonbufsize	<p>目的: First Failure Data Capture (FFDC) にシステム・トレース情報を提供する Lightweight Memory Trace (LMT) の共通イベント用メモリー・トレース・バッファー・サイズを指定します。</p> <p>チューニング: デフォルト値は、参照システム全体のアクティビティー、ハードウェアおよびシステム特性のもとでのデータ生成に基づいています。この範囲の上限はハードウェアおよびシステム特性に基づいており、LMT リソースを共有するため mtrc_rarebufsize の現在値によって決まります。記録されたイベントは、システム・ダンプに保存されるか、またはユーザー・コマンドを介して報告されず (あるいは、この両方が行われます)。</p>
mtrc_enabled	<p>目的: Lightweight Memory Trace (LMT) 状態を定義します。</p> <p>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。有効にするには、どのような状態の変更でも、その後で bosboot およびシステム・リブートが必要です。</p>
mtrc_rarebufsize	<p>目的: First Failure Data Capture (FFDC) にシステム・トレース情報を提供する Lightweight Memory Trace (LMT) の希少イベント用メモリー・トレース・バッファー・サイズを指定します。</p> <p>チューニング: デフォルト値は、参照システム全体のアクティビティー、ハードウェアおよびシステム特性のもとでのデータ生成に基づいています。この範囲の上限はハードウェアおよびシステム特性に基づいており、LMT リソースを共有するため mtrace_commonbufsize の現在値によって決まります。記録されたイベントは、システム・ダンプに保存されるか、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、この両方が行われます)。</p>

項目	説明
tprof_cyc_mult	<p>目的: トレース・サンプリング頻度を制御する方法としてパフォーマンス・モニター PM_CYC およびソフトウェア・イベント・サンプリング頻度乗数を指定します。</p>
tprof_evt_mult	<p>目的: トレース・サンプリング頻度を制御する方法としてパフォーマンス・モニター PM_* イベント・サンプリング頻度乗数を指定します。</p>
tprof_inst_threshold	<p>目的: トレース・サンプリング頻度を制御する方法としてパフォーマンス・モニター・イベント・サンプル間の完了した命令の最小数を指定します。</p> <p>値:</p> <ul style="list-style-type: none"> • デフォルト: 1000 • 範囲: 1 から 2G-1 • タイプ: 動的 <p>診断: 適用なし</p> <p>チューニング: 適用なし</p>
tprof_evt_system	<p>目的: 非特権ユーザーがシステム全体にわたるパフォーマンス・モニター・イベント・サンプリングを使用することを許可または制限します。</p> <p>値:</p> <ul style="list-style-type: none"> • デフォルト : 0 • 範囲: 0、1 • タイプ: 動的 • 単位: ブール値 <p>チューニング:</p> <p>tprof_evt_system を使用可能 (値 1) にすると、非特権ユーザーは tprof コマンドおよび pmctl コマンドを使用して、システム全体にわたるパフォーマンス・モニター・イベント・サンプリングを実行できます。使用不可 (値 0) にすると、非特権ユーザーは tprof コマンドおよび pmctl コマンドの -y オプション指定で開始されたプロセスについてイベント・サンプリングを実行できます。使用不可モードでは、非特権ユーザーは、カーネルおよびカーネル・エクステンションのイベント・サンプリングを実行できません。</p>

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・パラメーターのリブート・レベルをリストするには、次のように入力します。

```
raso -r -a
```

関連資料:

72 ページの『nfsd コマンド』

関連情報:

ioo コマンド

schedo コマンド

tunchange コマンド

vmc コマンド

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

項目	説明
-y <i>template-file</i>	/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

- 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
```

tfiler 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 <tfiler 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、**tfiler** ファイルに以下を入力します。

```
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。

rbacqry コマンド

目的

プロセスの使用特権および権限セットをレポートします。

構文

```
/usr/sbin/rbacqry [-T | -C] -n programname [ -i auditfile] -u username [-t timeperiod]
```

```
/usr/sbin/rbacqry -c [-s] -u username -S
```

説明

rbacqry コマンドは、アプリケーションでロール・ベースのアクセス制御 (RBAC) を使用可能にするためのモニター・ユーティリティとして使用されます。**rbacqry** コマンドは、プログラムの実行後にそのプログラムによって使用された特権および権限をレポートします。これは監査サブシステムを利用して、プログラムおよびその `spawn` プロセスによって作成されたすべてのプロセスの特権および権限を記録します。

rbacqry コマンドは、システムが拡張 RBAC モードで実行している場合に作動します。このレポートから入手した特権を、**setsecattr** コマンドを使用してアプリケーションの `innateprivs` および `inheritprivs` 属性に割り当てることができ、このコマンドが RBAC で使用可能になります。RBAC を使用可能にするために、プロセスの子の特権を統合して `inheritprivs` 属性で指定するか、`/etc/security/privcmds` ファイルで子のために別個のエントリを指定することができます。

注:

- **rbacqry** コマンドは、AIX の監査サブシステムによって生成される監査レポートに依存しています。
- **rbacqry -c** コマンドが実行されると、`rbac` 監査クラスが `/etc/security/audit/config` ファイルに追加されます。監査クラスを手動で構成することができます。
- このユーティリティを使用して特権と許可をトレースする場合、大容量の監査ログが作成されるのを避けるために、`/etc/security/audit/config` ファイルで `rbac` 監査クラスを特定のユーザーに割り当ててください。
- **rbacqry** コマンドは、出力の一部として RBAC ロールを示唆または明示することはありません。指定されたプログラムによって使用された特権および権限のみを示します。
- **rbacqry** ツールを使用してシェル・スクリプトをトレースする場合、トレースするスクリプトの先頭行でシェル・インタープリター (例えば、`#!/usr/bin/ksh`) を指定する必要があります。

フラグ

項目	説明
<code>-c</code>	指定されたユーザーについて、 <code>/etc/security/audit/config</code> ファイルを <code>rbac</code> クラスで構成します。
<code>-C</code>	プロセス・ツリーの使用特権および権限セットを、コンマで区切られたリストで示します。このオプションを <code>-T</code> オプションと同時に使用することはできません。
<code>-i auditfile</code>	rbacqry コマンドによって処理する監査証跡ファイルを指定します。これを指定しない場合、フラグはデフォルトで <code>/audit/trail</code> ファイルを使用します。
<code>-n programname</code>	使用特権をトレースする必要のあるターゲット・プログラム名を指定します。
<code>-s</code>	監査サブシステムがオフになっている場合、このサブシステムを始動します。監査サブシステムがすでにオンになっている場合には、このサブシステムを再始動します。
<code>-S</code>	スタンザ・フォーマットで出力を表示します。
<code>-T</code>	プロセスの使用特権および権限セットをツリー・フォーマットで指定します。
<code>-t timeperiod</code>	現在のシステム日付を基準に、使用特権レポートの生成を開始する時点からの日数を指定します。
<code>-u username</code>	ユーザー名を指定します。このオプションは、そのユーザーに監査イベントを構成するために、またそのユーザーによって実行されたプロセスを照会するために必要です。

終了状況

エラー値
= 0
> 0

ディスクリプター
正常終了。
エラー。

セキュリティ

Trusted AIX システムでは、権限のあるユーザーだけが **restore** コマンドを実行できます。

項目	ディスクリプター
aix.fs.manage.restore	このコマンドの実行に必要です。

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権命令を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権についての詳細情報は、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

1. プログラムによって使用された特権と許可を判別するには、以下のいずれかの方法を使用します。

a. **rbacqry** コマンドの実行対象とする必要があるプログラムまたはアプリケーションが root 以外のユーザーによって実行される場合、以下のステップを実行します。

1) 次のように **setsecattr** コマンドを実行して、root ユーザーまたは権限のあるユーザーのシェルで、プログラムを RBAC に対して一時的に使用可能にします。

```
setsecattr -c accessauths=ALLOW_ALL innateprivs=PV_ROOT secflags=FSF_EPS progpath
```

注: プログラムのパスは、**rbacqry** コマンドが実行されるプログラムの絶対パス名でなければなりません。

2) **setkst -t cmd** を (root ユーザーまたは権限のあるユーザーとして) 実行して、上記の変更を有効にします。

3) root ユーザーまたは権限のあるユーザーのシェルで **rbacqry** コマンドを実行して、ユーザーを監査に対して構成します。

```
rbacqry -c -s -u username
```

4) root 以外のユーザーとして、指定されたプログラムまたはアプリケーションを実行します。

5) プログラム実行が完了した後、root ユーザーまたは権限のあるユーザーのシェルで **rbacqry** コマンドを実行して、使用された特権と権限を収集します。

```
rbacqry -n program -u username (additional options can also be used)
```

6) root ユーザーまたは権限のあるユーザーとして次のコマンドを実行し、ステップ (i) で追加されたプログラム・エントリを `/etc/security/privcmds` ファイルから削除します。

```
rmsecattr -c progpath; setkst -t cmd
```

b. **rbacqry** コマンドが実行される必要があるプログラムまたはアプリケーションが (root ログインとして、または **su** コマンドを使用した root への切り替えにより) root ユーザーによって実行される場合、以下のステップを実行します。

1) root ユーザーまたは権限のあるユーザーのシェルで **rbacqry** コマンドを実行して、ユーザーを監査に対して構成します。

```
rbacqry -c -s -u root
```

2) root ユーザーとして、指定されたプログラムまたはアプリケーションを実行します。

- 3) プログラム実行が完了した後、root ユーザーまたは権限のあるユーザーのシェルで **rbacqry** コマンドを実行して、使用された特権と権限を収集します。

```
rbacqry -n program -u root (additional options can also be used)
```

注: ステップ i および ii の後で **su** コマンドを使用して root ユーザーに切り替えることによって実行されたプログラムまたはアプリケーションをトレースする場合、次のように **rbacqry** コマンドを実行します。

```
rbacqry -n program -u user_name (additional options can also be used)
```

2. (aix 許可を持つユーザー Scooby によって実行された) **chfs** コマンドとその spawn プロセスによって使用された特権と許可をツリー・フォーマットの出力で判別するには、次のコマンドを実行します。

```
# rbacqry -n chfs -u scooby -T
```

CMD	AUTHORIZATIONS	USED_PRIVS
chfs	aix.fs.manage.change	PV_FS_RESIZE
%extendlv	aix.lvm.manage.extend	PV_AU_ADMIN PV_KER_ACCT
%putlvcb	aix.lvm.manage	PV_FS_MKNOD PV_KER_LVM PV_PROC_PRIV PV_DEV_QUERY
%lextendlv	aix.lvm.manage.extend	PV_AU_ADD PV_FS_MKNOD PV_KER_ACCT PV_DEV_QUERY PV_AU_PROC PV_PROC_PRIV PV_KER_LVM PV_SU_UID
%savebase	aix.system.boot.create	PV_AU_PROC PV_PROC_PRIV PV_KER_LVM PV_SU_UID PV_FS_MKNOD PV_KER_ACCT PV_DEV_QUERY
%compress	aix.fs.manage.backup	PV_KER_ACCT PV_SU_UID
.....		

3. (aix 許可を持つユーザー Scooby によって実行された) **chfs** コマンドによって使用された特権と許可を別の監査証跡ファイルから表示するには、次のコマンドを実行します。

```
# rbacqry -u scooby -n chfs -i /audit/trail_example
```

CMD	AUTHORIZATIONS	USED_PRIVS
chfs	Used_Auth: aix.fs.manage.change	PV_DAC_0 PV_FS_CHOWN PV_FS_RESIZE
	Checked_Auths:	

4. (aix 許可を持つユーザー Scooby によって実行された) **chfs** コマンドによって使用された特権のコンマ区切りリストを取得するには、次のコマンドを実行します。

```
# rbacqry -n chfs -u scooby -C
```

CMD	AUTHORIZATIONS	USED_PRIVS
chfs	aix.fs.manage.change	PV_FS_RESIZE
extendlv	aix.lvm.manage.extend	PV_AU_ADMIN,PV_KER_ACCT
putlvcb	aix.lvm.manage	PV_FS_MKNOD,PV_PROC_PRIV,PV_KER_LVM,PV_DEV_QUERY
lextendlv	aix.lvm.manage.extend	

```

PV_AU_ADD,PV_AU_PROC,PV_FS_MKNOD,PV_PROC_PRIV,
PV_KER_ACCT,PV_KER_LVM,PV_DEV_QUERY,PV_SU_UID
savebase          aix.system.boot.create
PV_AU_PROC,PV_FS_MKNOD,PV_PROC_PRIV,PV_KER_ACCT,
compress          aix.fs.manage.backup    PV_KER_LVM,PV_DEV_QUERY,PV_SU_UID
PV_KER_ACCT,PV_SU_UID
.....

```

この出力フォーマットは、USED PRIVS セットが `/etc/security/privcmds` データベース内の特権コマンドに追加される場合に役立ちます。

注: システム許可とカスタム許可をトレースできます。システム許可を出力に表示する必要がある場合、より高い許可 (例えば、aix 許可) をユーザーに割り当てる必要があります。

5. 監査のためにユーザー `scooby` を構成するには、次のコマンドを実行します。

- a. ユーザーを構成して、そのユーザーの監査を開始するには、次のコマンドを実行します。

```
#/usr/sbin/rbacqry -c -s -u scooby
```

サブシステムの監査が開始されます。

- b. 監査を再開せずに、監査のためにユーザーを構成するには、次のコマンドを実行します。

```
#/usr/sbin/rbacqry -c -u scooby
```

注: 監査が再開されないため、ユーザー `scooby` は監査サブシステムによってトレースされません。`scooby` 用のエントリが `/etc/security/audit/config` ファイルに作成されます。監査によってユーザーがトレースされるようにするには、監査サブシステムを手動で再始動する必要があります。あるいは、次のように、**rbacqry** コマンドを実行する必要があります。

```
#/usr/sbin/rbacqry -c -s -u scooby
```

ユーザー `scooby` は、既に監査のために構成されています。サブシステムの監査が開始されます。

6. 次のような **-S** フォーマットのスタンザを表示するには、以下のコマンドを使用します。

```
# rbacqry -u scooby -n chfs -S chfs:
Used_Auth=aix.fs.manage.change
Checked_Auths=
Used_Privs=PV_DAC_0,PV_FS_CHOWN,PV_FS_RESIZE
```

7. フォーマット・オプションを指定せずに **rbacqry** コマンドを実行するには、以下のコマンドを実行します。

```
# rbacqry -u scooby -n chfs
CMD          AUTHORIZATIONS          USED_PRIVS
-----
chfs         Used_Auth:                PV_DAC_0          PV_FS_CHOWN
             aix.fs.manage.change     PV_FS_RESIZE
             Checked_Auths:
```

注: *checked Auths* パラメーターが存在しない場合、*checked_Auths* パラメーターはブランクです。そうでない場合、**rbacqry** コマンドは、次のような *checked_auths* パラメーターを表示します。

```
# rbacqry -u scooby -n lsuser
CMD          AUTHORIZATIONS          USED_PRIVS
-----
lsuser      Used_Auth:                PV_AZ_CHECK      PV_DAC_R
             ALLOW_ALL                PV_DAC_X
             Checked_Auths:
             aix.security.user.list
             aix.security.user.audit
             aix.security.efs
```

ファイル

ファイル・パス
/audit/trail

説明
監査ログを取り込む監査ファイルを指定します。

関連資料:

764 ページの『**rmauth** コマンド』

878 ページの『**rolerpt** コマンド』

関連情報:

authrpt コマンド

lssecattr コマンド

特権コマンド・データベース

rbactoldif コマンド

目的

ローカルで定義された特定の役割ベースのアクセス制御 (RBAC) およびドメインの役割ベースのアクセス制御テーブルを LDIF フォーマットで標準出力 (**stdout**) に出力します。

構文

```
rbactoldif -d baseDN [ -s tables ]
```

説明

rbactoldif コマンドは、ローカルに定義された RBAC テーブルからデータを読み取り、結果を LDIF フォーマットで **stdout** に出力します。ファイルにリダイレクトされる場合は、**ldapadd** コマンドまたは **ldif2db** コマンドを使用して結果を LDAP サーバーに追加できます。

rbactoldif コマンドは **/etc/security/ldap/sectoldif.cfg** ファイルを読み取って、権限、ロール、特権コマンド、特権デバイス、およびデータのエクスポート先の特権ファイル・サブツリーにどのような名前を付けるかを決定します。 **rbactoldif** コマンドは、ファイルで定義されている AUTHORIZATION、ROLE、PRIVCMD、PRIVDEV、および PRIVFILE の各タイプにのみデータをエクスポートします。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識別名 (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、「ファイル参照」の **/etc/security/ldap/sectoldif.cfg** ファイルを参照してください。

フラグ

項目
-d baseDN

説明
RBAC データが下に置かれる指定 DN を指定します。

項目	説明
<code>-s tables</code>	読み取りの対象となるテーブルのセットを指定します。 <code>-s</code> フラグを指定しない場合は、すべての RBAC およびドメインの RBAC テーブルが読み取られます。以下の文字の 1 つ以上を指定します。これらの文字はそれぞれテーブル名を表します。
a	権限テーブルを指定します。
c	特権コマンド・テーブルを指定します。
d	特権デバイス・テーブルを指定します。
e	ドメイン・テーブルを指定します。
f	特権ファイル・テーブルを指定します。
o	ドメイン・オブジェクト・テーブルを指定します。
r	ロール・テーブルを指定します。
t	trvi テーブルを指定します。

セキュリティ

rbactoldif コマンドは、モード・ビット 500 を持ち、root グループおよびセキュリティ・グループにより所有されます。

アクセスされるファイル

ファイル	モード
<code>/etc/security/authorizations</code>	r
<code>/etc/security/roles</code>	r
<code>/etc/security/privcmds</code>	r
<code>/etc/security/privdevs</code>	r
<code>/etc/security/privfiles</code>	r
<code>/etc/security/.rbac_ids</code>	r
<code>/etc/security/domains</code>	r
<code>/etc/security/domobjs</code>	r

例

- すべての RBAC およびドメイン RBAC テーブルを基本 DN `cn=aixdata` の LDIF フォーマットにエクスポートするには、次のコマンドを使用します。

```
rbactoldif -d cn=aixdata
```

- 権限テーブルおよびロール・テーブルのみを基本 DN `cn=aixdata` にエクスポートするには、次のコマンドを入力します。

```
rbactoldif -d cn=aixdata -s ar
```

- `domobjs` テーブルのみを基本 DN `cn=aixdata` にエクスポートするには、次のコマンドを入力します。

```
rbactoldif -d cn=aixdata -s o
```

関連資料:

172 ページの『`nistoldif` コマンド』

関連情報:

`mksecdap` コマンド

`sectoldif` コマンド

`/etc/security/ldap/sectoldif.cfg` 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ーを持っています。**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

項目	説明
-F	IPv6 の転送を使用不可にします。
-H	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。
-N	ndpd-router デーモンを停止します。
-S	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

1. 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
```
2. 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
```
3. 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
```
4. 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

項目	説明
<code>/etc/rc.mobip6</code>	rc.mobip6 コマンドが入っています。

関連資料:

17 ページの『**ndpd-router** デーモン』

関連情報:

モバイル IPv6

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 POver 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	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	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	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はない限り、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	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予期されています。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。
5, 7	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 rc.powerfail によっては処理されません。このような状態で rc.powerfail 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/scripts/epow** 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

項目	説明
-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。

項目	説明
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

項目	説明
html	

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

rc.wpars コマンド

目的

`workload partition`を自動的に開始します。

構文

`/etc/rc.wpars`

説明

`/etc/rc.wpars` コマンドは、`autostart` option (`mkwpar/chwpar -A`) が使用可能になっているすべての `workload partitions` 上で `startwpar` コマンドを呼び出します。 `/etc/rc.wpars` コマンドは、システムが起動されるたびに自動的に実行されます。

関連資料:

676 ページの『`rebootwpar` コマンド』

関連情報:

`chwpar` コマンド

`devexports` コマンド

`lswpar` コマンド

`startwpar` 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 } { 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、**¥** (円記号)、**"** (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

項目	説明
-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 -f	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。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。このような目的では、レルムは DCE セルと同義です。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。

パラメーター

項目	説明
<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。 注: rcp コマンドでは : (コロン) がホスト名の終わりを表すものであると想定されるため、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 ¥ (円記号) を挿入しなければなりません。
<i>User@Host:File</i>	rcp 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、 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。
ファイル <i>Host:Directory</i>	ローカル宛先ファイルのファイル名を指定します。 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。 注: rcp コマンドでは : (コロン) がホスト名の終わりを表すものであると想定されるため、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 ¥ (円記号) を挿入しなければなりません。

項目	説明
<code>User@Host:Directory</code>	rcp 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 (<code>User@</code>)、ホスト名 (<code>Host</code>)、およびディレクトリー名 (<code>Directory</code>) 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、 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。
<code>Directory</code>	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。

項目	説明
0	正常終了。
>0	エラーが発生しました。

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、**.klogin** への読み取りおよび書き込みアクセス (アクセス権 = 600) は AIX 所有者 ID だけが持つようにする必要があります。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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。

- 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。

- 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。

- 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/rcmdtkt$$; ¥  
/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

項目	説明
\$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/rcp	SP Kerberos 4 rcp ルーチンを (適用できる場合) 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、「IBMAIX 並列システム・サポート・プログラム (PSSP) 管理ガイド」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 でアクセスできます。

Kerberos に関する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 でアクセスできます。

関連情報:

cp コマンド

kvalid ユーザー

umask コマンド

krshd コマンド

通信およびネットワーク

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

項目	説明
-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

項目	説明
<code>\$HOME/.maildelivery</code>	ローカル・メール送達のための MH 命令を提供します。
<code>\$HOME/.forward</code>	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

sendmail コマンド

slocal コマンド

whom コマンド

MH の `.maildelivery` ファイル

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 **\$HOME/.maildelivery** ファイルの中に置かれ、これは **rcvpack** コマンドをすべての着信メッセージで実行します。

フラグ

項目	説明
-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

項目	説明
\$HOME/ html	

関連資料:

306 ページの『**packf** コマンド』

655 ページの『**rcvtty** コマンド』

関連情報:

inc コマンド

sendmail コマンド

MH の **.maildelivery** ファイル

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、**\$HOME/.maildelivery** ファイルの中に置かれます。

rcvstore コマンド・フラグを、**\$HOME/.mh_profile** ファイル内に指定することができます。

フラグ

項目	説明
-create	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。
+Folder	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、 +inbox 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。
-nocreate	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。
-nopublic	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。 -nopublic 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。
-nozero	rcvstore 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。
-public	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。 -public 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。
-sequence Name	組み込まれたメッセージを <i>Name</i> 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。
-zero	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

項目	説明
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。
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:	<i>UserMHDiretory</i> (ユーザーの MH ディレクトリー) 変数を指定します。
Unseen-Sequence:	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。
Rcvstore:	rcvstore プログラムのフラグを指定します。

ファイル

項目	説明
\$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。

関連資料:

653 ページの『**rcvdist** コマンド』

関連情報:

inc コマンド

slocal コマンド

.mh_alias コマンド

MH の **.maildelivery** ファイル

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、**.maildelivery** ファイルの中に置かれます。

フラグ

項目	説明
-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

項目	説明
\$HOME/\$HOME/mh_profile	MH ユーザー・プロファイルが入っています。

関連資料:

653 ページの『**rcvdist** コマンド』

653 ページの『**rcvpack** コマンド』

関連情報:

sendmail コマンド

slocal コマンド

MH の **.maildelivery** ファイル

rdist コマンド

本書では、**/usr/bin/rdist** ファイルにある古い AIX **rdist** コマンドと、新しい **rdistd** デーモンと一緒に使用される新しい **/usr/sbin/rdist** コマンドについて説明します。

/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 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・ラインの引数
- 小型の配布ファイルとして機能するコマンド・ラインの引数

-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・ラインに **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。詳細については、『TCP/IP のファイル・フォーマット』 (ファイル参照 の) 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

項目	説明
-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。 名前 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。

項目	説明
-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。
-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。
-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。

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タリーは、以下のいずれかのフォーマットを取ります。

項目	説明
<i>VariableName</i> = <i>NameList</i>	配布ファイルのほかのエンタリーに使用する変数 (<i>SourceList</i> 、 <i>DestinationList</i> 、または <i>SubcommandList</i>) を定義します。
[<i>Label</i>] <i>SourceList</i> -> <i>DestinationList</i> <i>SubcommandList</i>	rdist コマンドに対して、 <i>SourceList</i> 変数で指定したファイルを <i>DestinationList</i> 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。
[<i>Label</i>] <i>SourceList</i> :: <i>TimeStampFile</i> <i>SubcommandList</i>	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。 <i>SourceList</i> 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。

エンタリー

項目	説明
<i>VariableName</i>	配布ファイル内で使用する変数を識別します。
<i>NameList</i>	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。
<i>SourceList</i>	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。
<i>DestinationList</i>	ホストにファイルのコピーを受信するよう指示します。
<i>SubcommandList</i>	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。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、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、**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 ¥ (円記号) 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。

項目	説明
install <i>Options</i> [<i>OptionalDestName</i>];	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。 <i>Options</i> 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、 <i>SourceList</i> 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。 <i>OptionalDestName</i> パラメーターはファイル名を変更します。
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 <i>login@host</i> でなければ、ローカル・ホストと同じです。	
notify <i>NameList</i> ;	更新ファイル、およびリストされた名前 (<i>NameList</i> 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 (<i>name@host</i>)。
except <i>NameList</i> ;	rdist コマンドは、 <i>NameList</i> 変数で指定したファイルを除き、 <i>SourceList</i> 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 <i>NameList</i> ;	rdist コマンドが、 <i>NameList</i> 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 <i>NameList</i> "String";	<i>NameList</i> 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 (" <i>String</i> " 変数) を指定します。 <i>NameList</i> 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 " <i>String</i> " 変数を実行する前に、現行ファイル名に設定されます。" <i>String</i> " 値は、" " (二重引用符) で囲む必要があり、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。

項目	説明
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%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、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

項目	説明
<code>/usr/bin/rdist</code>	<code>rdist</code> コマンドが入っています。
<code>\$HOME/distfile</code>	<code>rdist</code> 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。
<code>/tmp/rdist</code>	更新リストが入っています。これは一時ファイルです。

`/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` コマンドと、新しい `rdistd` デーモンと一緒に使用される新しい `/usr/sbin/rdist` コマンドについて説明します。

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 [ -F n ] [ -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・ライン・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・ホストをアクセスします。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・ライン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・ライン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。

項目	説明
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

項目	説明
-A num	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。
-a num	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。
-d var = value	<i>value</i> を変数 <i>var</i> に割り当てます。このオプションは、 distfile 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。
-F	子プロセスを fork せずに、すべてのクライアントを順番に更新します。
-f distfile	distfile を配布ファイルとして使用します。 distfile が - と指定されている場合は、標準入力から読み取ります。
-l logopts	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、メッセージ・ログのセクションを参照してください。
-L logopts	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 (rdistd) 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、メッセージ・ログのセクションを参照してください。
-M num	同時に実行する子 rdist 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。
-m machine	ファイルの更新を指定したマシンに制限します。複数の -m 引数を指定し、更新を、 distfile 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	コマンドを表示しますが、実行はしません。 -n フラグを使用して distfile 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。 <i>distopts</i> は、以下にリストされたオプションをコマンドで区切ったリストです。 <i>distopts</i> 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のみ設定されます。
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のみ設定されます。
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のみ設定されます。
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制します。
削除	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。

項目	説明
-o <i>distopts</i>	(<i>dist</i> 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たんになりません。例えば、 <i>/path/dir1/f1</i> および <i>/path/dir2/f2</i> のファイルのリストを <i>/tmp/dir</i> に <i>rdist</i> すると、 <i>/tmp/dir/dir1/f1</i> と <i>/tmp/dir/dir2/f2</i> ではなく、ファイル <i>/tmp/dir/path/dir1/f1</i> と <i>/tmp/dir/path/dir2/f2</i> が作成されます。
younger	ファイルは通常、 <i>mtime</i> と <i>size</i> 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 <i>rdist</i> 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。
-p < <i>rdist-path</i> >	ターゲット・ホスト上の指定されたパス内で <i>rdistd</i> サーバーを探します。
-P < <i>rdist-path</i> >	<i>transport-path</i> で指定されたトランスポート・プログラムを使用します。 <i>transport-path</i> は、可能なパ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、存在するパスの最初のコンポーネントが使用されます。
-t <i>timeout</i>	リモートの <i>rdist</i> サーバーからの応答を待つ <i>timeout</i> 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。

ファイル

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** 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。

変更 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 (通知)

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。

nerror

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。

fferror 致命的エラーのメッセージをログに記録します。

warning (警告)

nerror 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。

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。

エントリー

項目	説明
<i>VariableName</i>	配布ファイル内で使用する変数を識別します。
<i>NameList</i>	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。
<i>SourceList</i>	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。
<i>DestinationList</i>	ホストにファイルのコピーを受信するよう指示します。
<i>SubcommandList</i>	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。

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、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、**csb** 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 ¥ (円記号) を付けます。 **rdist** コマンドは、**csb** 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。

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。

notify NameList;

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、現行ファイル名に設定されます。

変数 *REMFIL*E には更新されたばかりのリモート・ファイルの絶対パス名が含まれ、変数 *BASEFIL*E には更新されたばかりのリモート・ファイルのベース名が含まれます。

"*String*" 値は二重引用符で囲む必要があり、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。

項目	説明
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。

- 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。

- /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>}
```

- .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

- 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。

- [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。

- 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

項目	説明
<code>/usr/sbin/rdist</code>	バージョン 6.1.5 の rdist コマンドが入っています。
<code>distfile</code>	入力コマンドが入っています。
<code>\$ TMPDIR/rdist*</code>	更新リスト用の一時ファイルです。

関連資料:

『`rdistd` コマンド』

905 ページの『`rsh` または `remsh` コマンド』

関連情報:

`ssh` コマンド

`ksh` コマンド

`sh` コマンド

`.rhosts` コマンド

通信およびネットワーク

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

項目	説明
-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、以下の終了値を返します。

0 正常終了。

>0 エラーが発生しました。

ファイル

項目
/usr/sbin/rdistd
/usr/bin/rdistd

説明
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。

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、**-Level** および **FileSystem** パラメーターを指定して、バックアップしたいファイルを示します。**-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

項目	説明
-bBlocks	一回の出力操作で書き込むブロック数を指定します。 <i>Blocks</i> 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。 <i>Blocks</i> 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。 <i>Density</i> 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	<i>Machine</i> 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、 <i>Device</i> 変数をファイル名 (<i>/dev/rmt0</i> ファイルなど) として指定します。 <i>Device</i> 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。
-sSize	<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 -c 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、 rdump コマンドはテープが交換されるまで待ちます。
-u	<i>/etc/dumpdates</i> 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。
-w	現在使用不可。
-W	<i>/etc/dumpdates</i> ファイルで検出されたファイルシステムを表示します。
-Level	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。
-?	使用方法メッセージを表示します。

パラメーター

項目	説明
<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ーの名前を指定します。 rdump コマンドは、物理デバイス名として <i>/etc/filesystems</i> 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 (/) 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。

項目	説明
0	コマンドが正常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。

セキュリティ

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**Issecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

1. 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。

2. 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。

3. 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

項目	説明
/etc/dumpdates	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。
/etc/filesystems	ファイルシステムの情報が入っています。
/dev/rhd4	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。
/usr/sbin/rdump	rdump コマンドが入っています。

関連情報:

find コマンド

dumpdates コマンド

filesystems コマンド

rmt コマンド

ファイル・コマンド

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

項目	説明
-p	& (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 ¥ (円記号) 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 <i>n</i> 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。 <i>n</i> 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

項目	説明
<i>VariableName?Prompt</i>	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。 <i>Prompt</i> 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。
<i>VariableName...</i>	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。

項目	説明
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
```

- 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。

- 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。

- 入力行のコピーをコマンドとして履歴・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" がコマンドとして履歴・ファイルに保管されます。

関連資料:

456 ページの『**printf** コマンド』

関連情報:

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

項目	説明
-d <i>device</i>	読み取り元の論理ボリューム・デバイス・ファイル。
-c <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。
-C <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	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す。
-n <i>number_of_blocks</i>	読み取る 128K ブロックの数。
-o <i>outfile</i>	宛先ファイル。デフォルトは <i>stdout</i> です。
-s <i>skip</i>	<i>device</i> にスキップする 128K ブロックの数。
-S <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。autorestart 属性の値を見つけるには、**lsattr** コマンドを使用し、**lsattr -D -l sys0** と入力します。デフォルト値は **true** です。autorestart 属性の値を **false** にリセットするには、ログイン・アカウンティング・ファイルの */var/adm/wtmp* を使用します。**-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

項目	説明
-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。 -l フラグはアカウンティング・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。

項目 説明
-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。

*mmd**dHHMM* [*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)。2桁の値は現行の世紀の年数の値 (システム時刻に基づく) を表しています。例えば、システム時刻に基づく現行年が 1985 年である場合、99 は 1999 年を意味し、現行年が 2005 年である場合、99 は 2099 年を、04 は 2004 年を意味します。

セキュリティ

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
<i>/etc/rc</i>	システム始動スクリプトを指定します。
<i>/var/adm/wtmp</i>	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

fsck コマンド

lsattr コマンド

syslogd コマンド

utmp コマンド

rebootwpar コマンド

目的

システム・workload partitionを停止して再始動します。

制限: **rebootwpar** コマンドは、アプリケーション・workload partitionでは実行できません。

構文

```
rebootwpar [ -F | -h ] [ -N | -t seconds ] [ -v ] WparName
```

説明

rebootwpar コマンドは、workload partitionを停止して再始動します。

フラグ

項目	説明
-F	強制停止を指定します。
-h	ハード・ストップを指定します。
-N	停止のタイムアウトがないことを指定します。
-t seconds	停止のタイムアウト (秒) を指定します。
-v	冗長モード。
WparName	ワークロード・パーティション名を指定します。

セキュリティ

アクセス制御: root ユーザーだけがこのコマンドを実行できます。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**Issecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

「MyWpar」と呼ばれるworkload partitionをリブートするには、次のように入力します。

```
rebootwpar MyWpar
```

関連情報:

chwpar コマンド

clogin コマンド

startwpar コマンド

syncwpar コマンド

wparexec コマンド

recfgct コマンド

目的

Reliable Scalable Cluster Technology (RSCT) サブシステムを再構成します。

構文

```
/opt/rsct/install/bin/recfgct [ -i Node_ID | -n | -s | -h ]
```

説明

重要: このコマンドを使用する際には、細心の注意を払ってください。

recfgct コマンドを使用すると、**/var/ct** ディレクトリーの下にあるすべての RSCT データを削除し、新規 ノード ID を生成し、RSCT コンポーネントがインストールされたばかりのように見せることができます。このコマンドには破壊的な性質があるため、通常はシステム管理者がこれを始動することはありません。このコマンドは、重複したノード ID を除去する必要がある場合、または IBM サービス担当員から使用を指示された場合にのみ 使用してください。

RSCT が最初にインストールされると、ノード ID が自動的に生成されます。ノード ID は真のランダム 64 ビット数値です。RSCT がインストールされている各システムに、固有のノード ID が必要です。RSCT がインストールされているオペレーティング・システム・イメージ (OSI) のコピーが、別のシステムにインストールされた場合には、別のシステムもコピー元のシステムと同じノード ID を持つようになります。これはクローン作成と呼ばれます。AIX プラットフォームの場合、クローン作成は通常、**mksysb** のような AIX でサポートされているコマンドとプロシージャーによって実行されます。これらのコマンドおよびプロシージャーは自動的に **recfgct** を呼び出します。その他のプラットフォームの場合には、クローンの OSI がインストールされた直後に、**recfgct** コマンドを実行する必要があります。

-s フラグが指定されていると、**/var/ct** ディレクトリーの下でのデータがすべて除去された後、**/etc/ct_node_id** ファイルに含まれているノード ID を使用して **/var/ct/cfg/ct_node_id** ファイルが再作成されます。

フラグ

-i Node_ID

これは、使用しなければならないノード ID を指定します。このノード ID には、9 文字から 16 文字までの 16 進文字が含まれていなければなりません。

-n これは新規ノード ID を生成します。オプションが指定されていない場合、これがデフォルトの動作です。

-s ノード ID を保存します。

-h コマンド使用状況の説明文を標準出力に書き込んでから、終了します。

制約事項

-h フラグは、次の RSCT レベルでサポートされます。

- AIX 5.3 の場合は RSCT 2.4.9.1 (またはそれ以降)
- AIX 6.1 およびすべての Linux プラットフォームの場合は RSCT 2.5.1.1 (またはそれ以降)
- AIX 7.1 以降の場合は RSCT 3.1.0.0 (またはそれ以降)

recfgct -h コマンドをそれより前のバージョンの RSCT で実行しようとする、**-h** フラグが無視されて、すべての RSCT データが除去されます。

ファイル

/etc/ct_node_id

RSCT ノード ID のコピーが入っています。

/var/ct/cfg/ct_node_id

RSCT ノード ID が入っています。

標準出力

-h フラグが指定されている場合、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、次のコマンドが終了します。

終了状況

- 0 コマンドが正常に実行されました。
- 1 コマンドは正常に実行されませんでした。

セキュリティ

特権制御: このコマンドに対して実行 (x) 権限をもつのは、**root** ユーザーのみで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 オペレーティング・システムでは **rsct.core** ファイルセットの一部であり、Linux、Solaris、および Windows オペレーティング・システムでは **rsct.core-3.1.0.0-0.platform.rpm** パッケージの一部です。ここで *platform* は、**i386**、**ppc**、**ppc64**、**s390**、または **x86_64** です。

位置

/opt/rsct/install/bin/recfgct

例

1. クローンのオペレーティング・システム・イメージをインストールした後、次のように入力します。

```
/opt/rsct/install/bin/recfgct
```

関連情報:

who コマンド

recreatevg コマンド

目的

指定されたディスク・セット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 ] [ -YLv_Prefix | -l LvNameFile] [ -L Label_Prefix] [ -n ] [-V MajorNumber ] [ -d ] [ -O ] 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。

1. **recreatevg** コマンドは、コマンド・ライン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・ラインに指定されたディスクにファイルシステムの論理ボリュームのログ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. 同時可能のボリューム・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

項目	説明
-d	d フラグが指定された recreatevg コマンドは、VG を完全に作成し直す代わりに、指定のディスクに対して新規 PVID のみを作成し、その新規 PVID で LVM メタデータを更新します。論理ボリューム (LV) の名前とラベルは変更されません。また、VG はインポートされません。このフラグは、-0 フラグ以外の他のフラグとは一緒に指定できません。
-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になります。
-l <i>LVNameFile</i>	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。そのため、ディスクのサブセットにコピーの 1 つが存在する場合、より少ないミラー・コピーを使用して、ミラーリングされた LV を再作成できます。
-l <i>Label_Prefix</i>	論理ボリューム名を <i>LVNameFile</i> で指定された名前に変更します。エンタリーは LV:NEWLV1 フォーマットになっていなければなりません。 <i>LVNameFile</i> 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 <i>LVNameFile</i> スタンザ内の LV 名と同じにして (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 <i>Label_Prefix</i>	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、 <code>/etc/filesystems</code> ファイル・パスを手動で修正する必要があります。/ (スラッシュ) を <i>Label_Prefix</i> 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。
-0	ディスク上のメタデータにより、このボリューム・グループが別のノードでオンに変更されていることが示されている場合でも、ボリューム・グループを強制的に再作成してオンに変更します。詳細については、 varyonvg コマンドを参照してください。
-VMajorNumber	ボリューム・グループのメジャー番号を自動的に生成するのではなく、指定することができます。
-y <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。新しいボリューム・グループ名は、標準出力に送信されます。

項目
-Y *Lv_Prefix*

説明
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。デフォルトの名前は、以下の条件に従う必要があります。

- 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。
- 既に別のシステムによって使用されている名前を使用することはできません。

Lv_Prefix 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

1. 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、次のコマンドを入力します。

```
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。

2. 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、次のコマンドを入力します。

```
recreatevg -y testvg hdisk1
```

3. 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、次のコマンドを入力します。

```
recreatevg -Y newlv hdisk14
```

ファイル

項目	説明
/usr/sbin	recreatevg コマンドが存在するディレクトリ。

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** が表示されることとなります。 **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

項目	説明
/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:

bsh コマンド

ksh コマンド

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、あるいは **system** グループのメンバーでなければなりません。

フラグ

項目	説明
-d <i>Device</i>	ボリューム・グループ ID、 <i>Vgid</i> 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 <i>Vgid</i> でも指定できます。
-i <i>Vgid</i>	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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
<code>/usr/sbin/synclvodm</code>	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 コマンドを開始する前に、**rmlv** コマンドまたは **-d** フラグを使用して、*PhysicalVolume* パラメーターで示され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する必要があります。

注:

1. 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、あるいは **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。
4. **reducevg** コマンドは、アクティブなファームウェア支援ダンプ論理ボリュームを持つボリューム・グループには使用できません。
5. **reducevg** コマンドは、ボリューム・グループから除去された物理ボリュームに対して実行中のすべてのバックグラウンド・スペース再利用プロセスを破棄します。スペース再利用が実行中であるかどうかを識別するには、**lvmstat** コマンドに **-r** オプションを指定します。

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。

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

項目	説明
-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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
<code>/usr/sbin/reducevg</code>	<code>reducevg</code> コマンドが常駐するディレクトリー。
<code>/tmp</code>	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。

関連資料:

814 ページの『`rmlv` コマンド』

関連情報:

`extendvg` コマンド

論理ボリューム・ストレージ

System Management Interface Tool

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 **indxbib** 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 **indxbib** コマンドを再発行しなければなりません。**indxbib** 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

項目	説明
-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/Ind 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。

項目
-sKeys

説明
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。

```
%A M.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

項目
/usr/share/dict/papers/Ind
/usr/lbin/refer

説明
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。

関連資料:

24 ページの『**neqn** コマンド』

230 ページの『**nroff** コマンド』

874 ページの『**roffbib** コマンド』

関連情報:

メッセージ・カタログ

lookbib 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

項目	説明
-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。
+Folder	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。
-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。
-link <i>Messages</i>	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。 <i>Number</i> メッセージの数。 <i>Sequence</i> 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、 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。
-src +Folder	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。

項目	説明
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:	<i>UserMhDirectory</i> 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。

セキュリティ

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
<code>\$HOME/.mh_profile</code> <code>/usr/bin/refile</code>	MH ユーザー・プロファイルを設定します。 <code>refile</code> 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 コマンド
 .mh_profile コマンド
 メール・アプリケーション

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

項目	説明
-g <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・クラス内に入っていないと、 refresh コマンドは失敗に終わります。
-h <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" として実行中でなければなりません。リモート・システムは、リモート・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に構成する必要があります。すなわち、 <code>srcmstr</code> デーモン (<code>/etc/inittab</code> を参照) が -r フラグで始動され、 <code>/etc/hosts.equiv</code> または <code>.rhosts</code> ファイルがリモート要求が行えるように構成される必要があります。
-p <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。
-s <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と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、 refresh コマンドは失敗に終わります。

セキュリティー

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティー」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、`lssecattr` コマンドまたは `getcmdattr` サブコマンドの項を参照してください。

例

1. tcpiip 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 tcpiip
```

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

項目	説明
/etc/objrepos/SRCsubsys	SRC サブシステム構成オブジェクト・クラスを指定します。
/etc/services	インターネット・サービスに使われるソケットとプロトコルを定義します。
/dev/SRC	AF_UNIX ソケット・ファイルを指定します。
/dev.SRC-unix	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

システム・リソース・コントローラー

refrsrc 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

```
refrsrc [-h] [-TV] resource_class
```

説明

refrsrc 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 (Resource Monitoring and Control) 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・ライン・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・ラインに間違ったフラグが指定されました。
- 4 コマンド・ライン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・ライン入力に基づく 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 (Resource Monitoring and Control) 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 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. 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。

```
refrsrc IBM.FileSystem
```

位置

`/opt/rsct/bin/refrsrc`

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またはマイクロセンサーをリフレッシュします。

構文

センサーをリフレッシュするには、次のように実行します。

```
refsensor [-a | -n host1[,host2...] | -N { node_file | "-" } ] [-h ] [ -v | -V ] sensor_name
```

マイクロセンサーをリフレッシュするには、次のように実行します。

```
refsensor -m [-a | -n host1[,host2...] | -N { node_file | "-" } ] [-h ] [ -v | -V ] sensor_name
```

説明

refsensor コマンドは、RMC サブシステムに対して定義されているセンサー・リソースまたはマイクロセンサー・リソースをリフレッシュします。センサーおよびマイクロセンサーは、モニターが可能な属性を含む RMC リソースです。**refsensor** が正常に実行されるためには、センサーとマイクロセンサーがモニター中でなければなりません。

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。マイクロセンサーは、**refsensor** を使用してマイクロセンサーのロード・モジュールの値を照会するとリフレッシュできます。**-m** フラグを使用してマイクロセンサーをリフレッシュします。

センサー・コマンドの実行またはマイクロセンサーのロード・モジュールの照会について間隔が (**mknsensor** コマンドを使用して) 定義されている場合、**refsensor** コマンドを実行しても、間隔には影響し

ません。つまり、モニター対象のセンサーまたはマイクロセンサーが 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** を実行したい場合は、**-n** フラグを使用します。**-n** フラグを使用して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内にあることを指示することができます。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。

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。

Float32

このセンサー・リソースに関連した **float32** タイプの属性。

Float64

このセンサー・リソースに関連した **float64** タイプの属性。

Int32 このセンサー・リソースに関連した **int32** タイプの属性。

Int64 このセンサー・リソースに関連した **int64** タイプの属性。

Quantum

このセンサー・リソースに関連した **quantum** タイプの属性。

string このセンサー・リソースに関連した **string** タイプの属性。

Uint32

このセンサー・リソースに関連した **uint32** タイプの属性。

Uint64

このセンサー・リソースに関連した **uint64** タイプの属性。

例えば、センサー名 **Sensor1** について、センサー属性 **Int32** および **Float32** を更新するには、次のように入力します。

```
refsensor Sensor1 Int32=45 Float32=7.8
```

マイクロセンサー属性は単独で更新することはできません。

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 **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。

-m リフレッシュするリソースがマイクロセンサー・リソースであると指定します。

-n *host1*[,*host2*...]

センサーをリフレッシュする 1 つ以上の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* | "-" }

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。
- ノード名の左の空白文字はすべて無視されます。
- ノード名の右の空白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、管理ドメインまたはピア・ドメインで **-N** "-" を使用します。

-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。

-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

このコマンドを使用してセンサーをリフレッシュするには、**IBM.Sensor** リソース・クラスに対する書き込み権限が必要です。

このコマンドを使用してマイクロセンサーをリフレッシュするには、**IBM.MicroSensor** リソース・クラスに対する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。
- 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 (Resource Monitoring and Control) 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 では **rsct.core** ファイルセットの一部であり、Linux、Solaris、および Windows では **rsct.core-3.1.0.0-platform.rpm** パッケージの一部です。ここで *platform* は、**i386**、**ppc**、**ppc64**、**s390**、または **x86_64** 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"
```

3. **/u/joe/common_nodes** ファイルにリストされているノード上のセンサー **Sensor1** をリフレッシュして、その **Sensor1** 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 -N /u/joe/common_nodes Sensor1
```

この **/u/joe/common_nodes** には、以下のものが含まれます。

```
# common node file
#
node1.myhost.com    main node
node2.myhost.com    backup node
```

4. **IBM.Sensor1** と呼ばれるマイクロセンサーをリフレッシュして、定義されているマイクロセンサーのロード・モジュールを使用して属性値が照会されるようにするには、次のように入力します。

```
refsensor -m IBM.Sensor1
```

位置

`/opt/rsct/bin/refsensor`

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* の各エントリーは、後に 1 つまたは複数のブランクと " " (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

項目 説明

- 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

項目	説明
<code>/usr/ccs/bin/regcmp</code>	<code>regcmp</code> コマンドが入っています。

関連情報:

`regcmp` コマンド

サブルーチンの概要

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・ライン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

項目	説明
<code>-#JobNumber</code>	取り消したい <i>JobNumber</i> を指定します。
<code>-C</code>	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 つのハンドシェイク・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。 <code>-C</code> 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。
<code>-D DebugOutputfile</code>	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。
<code>-L</code>	リモート・キューからの長い (詳細) 状況要求を示します。

項目	説明
-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、 <code>/etc/qconfig</code> ファイル内の <code>s_statfilter</code> 属性で行います。 <code>filter</code> 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 <i>Option</i> を指定します。(これらの <i>Options</i> 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 <i>Queue</i> 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 <i>Server</i> 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	<i>HostName</i> コンピューターから実行要求された <i>UserName</i> 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、 <i>Option</i> 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 <code>/etc/qconfig</code> ファイルに含めなければなりません。 <code>backend = /usr/lib/lpd/rembak -X</code>
-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。 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

- 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`
- 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`
- バージョン 3 のコンピューター `bluto` からユーザー `sweetpea` が実行要求したリモート・サーバー上のジョブ番号 23 を取り消すには、次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

項目	説明
<code>/usr/lib/lpd/rembak</code>	rembak コマンドが入っています。
<code>/etc/hosts.lpd</code>	印刷要求を実行できるホスト名が入っています。
<code>/etc/hosts.equiv</code>	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
disable コマンド
enable コマンド
印刷の管理
印刷プーラー

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

項目	説明
<code>/usr/sbin/acct</code>	アカウントティング・コマンドへのパス。
<code>/var/adm/acct/nite</code>	アカウントティング・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:

システム・アカウントティング
アカウントティング・サブシステムの設定

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、除去し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が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 **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。

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
```

rendev コマンド

目的

デバイスの名前を変更します。

構文

```
rendev -l Name -n NewName [-u]
```

説明

rendev コマンドにより、デバイスの名前を変更できます。名前変更を行うデバイスは **-l** フラグを用いて指定され、希望する新規名は **-n** フラグを用いて指定されます。

希望する新規名の長さは 15 文字を超えてはなりません。この名前がすでに使用されているかまたは `/dev` ディレクトリー内に存在する場合は、この操作は失敗します。文字 `r` の後に新規名を付加して形成される名前が、デバイス名としてすでに使用されている場合または `/dev` ディレクトリー内に存在する場合は、この操作は失敗します。

デバイスが「使用可能」状態の場合は、**rendev** コマンドにより、デバイスの名前を変更する前にそのデバイスを構成解除する必要があります。これは、**rmdev -l Name** コマンドによって実行される操作に類似しています。構成解除操作が失敗した場合は、名前変更も失敗します。構成解除が成功した場合は、**rendev** コマンドは、デバイスを「使用可能」状態に復元するために、デバイスを名前変更後に構成します。**-u** フラグを使用して、デバイスが名前変更後に再構成されないようにすることができます。

注: ルート・ボリューム・グループのメンバーであるか、ルート・ボリューム・グループのメンバーになる予定の (LVM またはインストール手順を使用して) ディスク・ドライブ・デバイスは名前変更できません。このようなディスク・ドライブを名前変更すると、ブートの失敗など特定のシナリオから復旧するための機能が失われる場合があります。

一部のデバイスは、他のデバイスまたはアプリケーションがその名前を使用するために、その名前に関する特殊な要件を持つ場合があります。このようなデバイスの名前を変更するために **rendev** コマンドを使用すると、そのデバイスが使用できなくなることがあります。

注: 構成データベースを保護するために、**rendev** コマンドは、開始後には中断できません。このコマンドを、完了しないうちに停止しようとする、データベースが破壊される可能性があります。

フラグ

項目	説明
<code>-l Name</code>	カスタマイズ・デバイス・オブジェクトで名前変更される、 <code>Name</code> パラメーターで示されたデバイスを指定します。
<code>-n NewName</code>	デバイスに割り当てられる、 <code>NewName</code> パラメーターで示された新規名を指定します。
<code>-u</code>	オプション・フラグ。これは、デバイスが名前変更後に構成されないことを示します。

例

1. ディスクの名前を `hdisk5` から `hdisk2` に変更するには、次のように入力します。

```
rendev -l hdisk5 -n hdisk2
```

2. ディスクの名前を `hdisk3` から `ootvg` に変更するには、次のように入力します。

```
rendev -l hdisk3 -n ootvg
```

2 番目のコマンドが失敗するのは、`r` に付加された `ootvg` により名前 `rootvg` が生じるためです。この名前は `rootvg` ボリューム・グループ名と競合します。

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。

項目	説明
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

項目	説明
-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。

項目	説明
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

項目	説明
/usr/sbin/renice	renice コマンドが入っています。
/etc/passwd	ユーザー名をユーザー ID にマップします。

関連資料:

86 ページの『nice コマンド』

関連情報:

getpriority コマンド、setpriority コマンド

processes コマンド

shells コマンド

マイクロプロセッサのコンテンションの制御

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

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** ユーザー権限を持っているか、あるいは **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。
6. **reorgvg** コマンドは、アクティブなファームウェア支援ダンプ論理ボリュームを持つボリューム・グループには使用できません。

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

項目	説明
-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。

セキュリティ

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
/usr/sbin/reorgvg	reorgvg コマンドが入っているディレクトリ。
/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。

関連情報:

chlv コマンド

lslv コマンド

System Management Interface Tool

論理ボリュームおよびディスク入出力のパフォーマンス

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 ] [ -draftmessageMessage ]
[ -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か、**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** コマンドを開始します。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、ダッシュ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

項目	説明
-annotate	応答中のメッセージに注釈として応答日時を付けます。 -inplace 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。
-cc Names	応答の cc: フィールドにリストされるユーザーを指定します。 Names には、 all 、 to 、 cc 、および me を指定できます。デフォルトは、 -cc all です。
-draftfolder +Folder	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。 +Folder を指定しないと、 Current-Folder が使用されます。
-draftmessage Message	ドラフト・メッセージを指定します。 -draftmessage フラグを付けずに -draftfolder を指定すると、デフォルト・メッセージは新規となります。 -draftfolder 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 UserMHdirectory/draft 内でドラフト・メッセージを作成します。
-editor Editor	応答を構成する最初のエディターを識別します。 -editor フラグを指定しないと、 comp コマンドは、 \$HOME/mh_profile 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。
-fcc +Folder	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、 repl コマンドはファイル・コピーを生成しません。
-filter File	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。 File 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。 -filter フラグは、 mhl コマンドで使えるフォーマット・ファイルを使用します。
+Folder	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。
-form FormFile	応答フォーマットを指定します。 repl 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。

項目	説明
-format	To:、cc:、および Bcc: フィールドから重複アドレスを除去し、 -width 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。 -format 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。
-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。
-inplace	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、 -draftmessage 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。 cur または . (ピリオド) 現行メッセージ。デフォルトの応答メッセージ。 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。 new 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 new です。 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。 注釈を付けません。このフラグがデフォルトです。 -nocc Names 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 all 、 to 、 cc 、および me を指定できます。 -nodraftfolder <i>UserMhDirectory/draft</i> 内にドラフトを配置します。 -noedit 初期編集を抑制します。 -noformat 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。 -notinplace 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。 -noquery 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。 -nowhatnowproc repl 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。 -query 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。 -whatnowproc Program 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、 whatnow プログラムです。 -width Number 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。

項目	説明
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。
mhlproc:	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。

セキュリティー

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティー」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

1. 現行フォルダーの現行メッセージに返信するには、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返信します。

```
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>
```

これで返信を入力できます。返信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返信します。

```
What now?
```

返信を送信するには、**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返信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 に返信するには、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返信します。

```
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>
```

これで返信を入力できます。返信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返信します。

```
What now?
```

返信を送信するには、**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返信を追跡するには、以下のように **-annotate** フラグを使用して、返信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返信します。

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

項目	説明
<code>\$HOME/mh_profile</code>	ユーザーの MH プロファイルを指定します。
<code>/etc/mh/replcomps</code>	MH のデフォルトの応答テンプレートが入っています。
<code>UserMhDirectory/replcomps</code>	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。
<code>/usr/bin/repl</code>	repl コマンドが入っています。
<code>UserMhDirectory/draft</code>	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
メール・アプリケーション

replacev 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

```
replacev [ -f ] {SourcePhysicalVolume | SourcePhysicalVolumeID } DestinationPhysicalVolume
```

```
replacev [ -R ] dir_name [ DestinationPhysicalVolume ]
```

説明

replacev 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* は、ボリューム・グループに属してはなりません。
2. *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。

3. **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。
4. **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。
5. アクティブなファームウェア支援ダンプ論理ボリュームを持つ物理ボリュームに対してこのコマンドを実行すると、ダンプ・デバイスが一時的に */dev/sysdumpnull* に変更されます。論理ボリュームのマイグレーションが正常に実行された後、このコマンドは **sysdumpdev -P** コマンドを呼び出して、ファームウェア支援ダンプ論理ボリュームを元の論理ボリュームに設定します。
6. *SourcePhysicalVolume* に対応する VG は、PV タイプの制限が存在するかどうかを判別するために検査されます。制限が存在する場合、*DestinationPhysicalVolume* は、その制限を確実に満たすように検査されます。PV タイプの制限を満たさない場合は、このコマンドは失敗します。

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

項目	説明
-f	<i>SourcePhysicalVolume</i> を、指定された <i>DestinationPhysicalVolume</i> と置き換えるように強制します。ただし、 <i>DestinationPhysicalVolume</i> 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、 <i>DestinationPhysicalVolume</i> 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

項目	説明
<i>/usr/sbin</i>	replacepv コマンドが常駐しているディレクトリー。
<i>/tmp</i>	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

論理ボリューム・ストレージ

System Management Interface Tool

システム・ダンプ機能

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

項目	説明
-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。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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			File limits		
		soft	hard	grace	used	soft	hard
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

項目	説明
quota.user	ユーザー・クォータを指定します。
quota.group	グループ・クォータを指定します。
/etc/filesystems	ファイルシステムの名前と位置が入っています。
/etc/group	基本グループ属性が入っています。
/etc/passwd	ユーザーの名前と位置が入っています。

関連資料:

- 619 ページの『quota コマンド』
- 623 ページの『quotaon または quotaoff コマンド』
- 621 ページの『quotacheck コマンド』

関連情報:

- ディスク・クォータ・システムの概要
- ディスク・クォータ・システムのセットアップ

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 ] [ -I ] [ -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

項目	説明
-	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。
-I	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。
-m Identifier TestbaudRate: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 パラメーターにポー・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、ポー・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 cs h コマンドのシーケンスを出力します。

ファイル

項目	説明
/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:

cs

sh コマンド

env

terminfo コマンド

TTY 端末デバイス

resetsrc コマンド

目的

リソースをリセットします。すなわち、リソースを強制的にオフライン状態に移動します。

構文

コマンド・ラインに入力したデータを使用して 1 つ以上のリソースをリセットするには、以下のように実行します。

```
resetsrc -s "selection_string" [ -N { node_file | "-" } ] [-h] [-TV] resource_class [arg=value...]
```

```
resetsrc -r [-h] [-TV] resource_handle [arg=value...]
```

入力ファイルに事前定義されたコマンド引数を使用して 1 つ以上のリソースをリセットするには、以下のように入力します。

```
resetsrc -f resource_data_input_file -s "selection_string" [ -N { node_file | "-" } ] [-h] [-TV] resource_class
```

```
resetsrc -f resource_data_input_file -r [-h] [-TV] resource_handle
```

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。

```
resetsrc -l [-h] resource_class
```

説明

resetsrc コマンドは、Resource Monitoring and Control (RMC) サブシステムに要求を出して、1 つ以上のリソースを強制的にオフラインにします。要求は該当するリソース・マネージャーによって実行されません。

1 つ以上のリソースをリセットするには、指定の選択文字列に一致するすべてのリソースを強制的にオフラインにする **-s** フラグを使用します。特定の 1 つのリソースをリセットするには、その特定のリソースを表すリソース・ハンドルを指定する **-r** フラグを使用します。

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。

指定したリソース・クラスが任意の追加のコマンド引数を受け入れるかどうかを判別するには、**-l** フラグを使用します。

このコマンドが正常終了しても、リソースがオフラインになることが保証されるわけではなく、リソース・マネージャーが、このリソースのオフラインを強制するための要求を正常に受け取ったにすぎません。リソースが強制的にオフラインになる時期については、リソースの動的属性 **OpState** をモニターして判別します。リソースにイベントを登録して、**OpState** 属性を指定し、リソースがいつオフラインになるかを識別します。あるいは、リソースがオフライン (**OpState** の値が **2**) であることが確認されるまで、**lsrsrc** コマンドを断続的に実行します。以下に例を示します。

```
lsrsrc -s 'Name == "/filesys1"' -t IBM.FileSystem Name OpState
```

システムにクラスター・システム・マネジメント (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。

パラメーター

resource_class

強制的にオフラインにしたいリソースを含むリソース・クラスの名前を指定します。

resource_handle

強制的にオフラインにしたいリソースに対応するリソース・ハンドルを指定します。**lsrsrc** コマンドを使用して、有効なリソース・ハンドルのリストを入手します。リソース・ハンドルは、二重引用符で囲む必要があります。例えば、次のとおりです。

```
"0x4017 0x0001 0x00000000 0x0069684c 0x0d4715b0 0xe9635f69"
```

arg=value...

コマンド引数の名前と値のペアを 1 つ以上指定します。

arg 引数の名前を指定します。

value この引数の値を指定します。値のデータ・タイプは引数のデータ・タイプの定義と一致しなければなりません。

コマンドの引数はオプションです。*arg=value* のペアを入力する場合は、指定のリソース・クラスに対してオフライン機能が定義された、*arg=value* ペアがコマンド引数ごとに 1 つ存在しなければなりません。

特定のリソース・クラスのコマンド引数の名前とデータ・タイプのリストを入手するには、**resetsrc -l** を使用します。

フラグ

-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。以下のファイルの内容が表示されます。

```
PersistentResourceArguments::  
argument1 = value1  
argument2 = value2
```

-l 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、オフライン要求に渡された追加の引数を受け入れます。このフラグを使用して定義済みのコマンド引数とコマンド引数値のデータ・タイプをリストします。

-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。
- ノード名の左の空白文字はすべて無視されます。
- ノード名の右の空白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。

-r 指定したリソース・ハンドルに一致する特定のリソースを強制的にオフラインにします。

-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。

-h コマンドの使用方法のステートメントを標準出力に書き込みます。

-T 標準エラーにコマンドのトレース・メッセージを書き込みます。これはソフトウェア・サービス組織専用です。

-V コマンドの詳細メッセージ (使用可能なメッセージがある場合) を標準出力に書き込みます。

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 (Resource Monitoring and Control) 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。コマンドによって表示または変更されるリソース・クラスやリソースは、接続が確立されたシステム上にあります。

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_CONTACT が IP アドレスに設定されている場合のみ、CT_IP_AUTHENT が意味を持ちます。CT_IP_AUTHENT はドメイン・ネーム・システム (DNS) サービスに依存しません。

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。

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** フラグが指定されている場合は、このコマンドの詳細メッセージ (使用可能なメッセージがある場合) 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。

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・ライン・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・ラインに間違ったフラグが指定されました。
- 4 コマンド・ライン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・ライン入力に基づく RMC により、エラーが発生しました。
- 6 指定された選択文字列に一致するリソースが見つかりませんでした。

セキュリティ

resetrsrc を実行するには、**resetrsrc** 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 ファイルおよびその変更方法について詳しくは、「*Administering RSCT*」ガイドを参照してください。

実装上の固有な条件

このコマンドは AIX では **rsct.core.rmc** ファイルセットの一部であり、Linux、Solaris、および Windows では **rsct.core-3.1.0.0-0.platform.rpm** パッケージの一部です。ここで *platform* は、**i386**、**ppc**、**ppc64**、**s390**、または **x86_64** です。

位置

/opt/rsct/bin/resetrsrc

例

ピア・ドメイン **foo** に、**nodeA**、**nodeB**、および **nodeC** という 3 つの定義済みのノードがあり、**nodeA** には **ent0** および **ent1** という 2 つのイーサネット・カードがあるものとします。

1. **nodeA** がオンラインであり、**ent0** (**nodeA** 上にある) もオンラインであるとします。**nodeA** 上の **ent0** を強制的にオフラインにするには、次のコマンドを **nodeA** で実行します。

```
resetrsrc -s 'Name == "ent0"' IBM.EthernetDevice
```

2. **nodeA** および **nodeB** がオンラインであり、**ent0** (**nodeA** 上にある) もオンラインであるとします。また、現在、ユーザーは **nodeB** にログオンしているとします。**nodeA** 上の **ent0** を強制的にオフラインにするには、次のコマンドを **nodeB** で実行します。

```
resetrsrc -s 'NodeName == "nodeA" AND Name == "ent0"' IBM.EthernetDevice
```

3. **nodeA** および **nodeB** がオンラインで、ファイルシステム **/fileysys1** が **nodeB** に定義され、マウントされているとします。**nodeB** 上の **/fileysys1** を強制的にオフラインにするには、次のコマンドを **nodeA** で実行します。

```
resetrsrc -s 'NodeName == "nodeB" AND Name == "/fileysys1"' IBM.FileSystem
```

4. **nodeA** 上の **ent0** のリソース・ハンドルが以下であるとします。

```
0x406b 0x0001 0x00000000 0x0069564c 0x0dc1f272 0xb9de145e
```

nodeA 上の **ent0** を強制的にオフラインにするには、次のコマンドを **nodeA** で実行します。

```
resetrsrc -r "0x406b 0x0001 0x00000000 0x0069564c 0x0dc1f272 0xb9de145e"
```

5. **nodeA** 上および **nodeB** 上で **ent0** をリセットするには、**/tmp/common/node_file** ファイルを使用して次のように実行します。


```
# common node file
#
nodeA
nodeB
#
```

次のように入力します。

```
resetsrc -s 'Name == "ent0"' -N /tmp/common/node_file ¥
IBM.EthernetDevice
```

関連資料:

720 ページの『resource_data_input 情報ファイル』

772 ページの『rmccli 情報ファイル』

関連情報:

lsrsrc コマンド

startsrc コマンド

stopsrc コマンド

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・ライン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、以下のように入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

項目	説明
-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。 <i>Rows</i> パラメーターおよび <i>Columns</i> 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

項目	説明
/etc/termcap	基本 termcap エントリ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

resource_data_input 情報ファイル

目的

リソース属性名や値などのリソース・クラス情報を Resource Monitoring and Control (RMC) コマンド・ライン・インターフェース (CLI) に渡すための入力ファイルの使用方法について説明します。

説明

大部分の RMC コマンドには **-f** フラグを使用することができ、リソースの永続属性値や他の情報を RMC CLI に渡したい場合に、このフラグでリソース・データの入力ファイルの名前を指定します。このフラグは、コマンド・ラインに関する情報の入力が煩雑すぎる場合や、タイプミスの可能性が高い場合に便利です。このファイルのデータは、リソースを定義したり、リソースまたはリソース・クラスの永続属性の値を変更したりする際に使用します。POSIX フォーマットで作成しなければならない、リソース・データ入力ファイルには、設定場所はありません。要件に応じて、一時ファイルと永続ファイルのどちらにもすることができます。

このファイルは、**-f** フラグを指定して出された場合の **chrsrc**、**mkrsrc**、**resetrsrc**、**rmrsrc**、**runact**、**startsrc**、および **stopsrc** の各コマンドで読み取られます。**lsactdef**、**lsrsrc**、および **lsrsrcdef** の各コマンドは、**-i** フラグを指定して出された場合にこのフォーマットでファイルを生成します。

関連するスタンザにリストされているデータのタイプを示すために、入力ファイルに次のようなキーワードが使用されます。

ResourceAction

アクションを開始するときのリソース・アクションのリソース・アクション要素名と値です。

runact コマンドがリソース・アクション要素を読み取ります。入力ファイルが **runact -c** で読み取られるときは、これらの要素は無視されます。

ResourceClassAction

クラス・アクションを開始するときのリソース・クラス・アクションのリソース・クラス・アクション要素名と値です。**runact** コマンドがリソース・アクション要素を読み取ります。

PersistentResourceArguments

リソース・コマンドの引数の名前と値で、これらを受け入れるコマンドで使用されます。そのコマンドとは、**mkrsrc**、**resetrsrc**、**rmrsrc**、**startsrc**、および **stopsrc** です。コマンド引数はオプションで、リソース・クラスによって定義されます。リソース・クラスのコマンドの引数を表示する場合は、これらのコマンドに **-l** オプションを指定します。

PersistentResourceAttributes

新規リソースを定義するか、あるいは既存のリソースの属性値を変更するために使用される、特定のリソース・クラスの 1 つ以上のリソースの永続属性名および値です。永続リソース属性は、**mkrsrc** コマンドおよび **chrsrc** コマンドによって読み取られます。**-c** フラグが指定された **chrsrc** コマンドによって入力ファイルが読み取られるときには、これらの属性は無視されます。

PersistentResourceClassAttributes

既存のリソース・クラスの属性値を変更するために使用されるリソース・クラスの永続属性名および値です。永続リソース・クラス属性が **chrsrc** コマンドで読み取られるのは、**-c** フラグが指定されている場合のみです。

通常、*resource_data_input* ファイルは、次のフォーマットのフラット・テキスト・ファイルです。太字の単語はリテラルです。1 つのコロン (:) が先行するテキストは任意のラベルで、このラベルには任意の英数字テキストを使用できます。

```
PersistentResourceAttributes::
# This is a comment
  label:
    AttrName1 = value
    AttrName2 = value
    AttrName3 = value
  another label:
    Name      = name
    NodeNumber = 1
:
::

PersistentResourceClassAttributes::
# This is a comment
  label:
    SomeSettableAttrName = value
    SomeOtherSettableAttrName = value
  :
:

PersistentResourceArguments::
# This is a comment
  label:
    ArgName1 = value
    ArgName2 = value
    ArgName3 = value
  :
:
```

詳細については、『例』のセクションを参照してください。

フォーマットに関する注の一部は次のとおりです。

- キーワード `PersistentResourceAttributes`、`PersistentResourceClassAttributes`、および `PersistentResourceArguments` には、2 つのコロン (:) が後続します。
- キーワード・スタンザの順序はこのファイルでは重要ではありません。例えば、`PersistentResourceClassAttributes` は、`PersistentResourceClass` の前に置くことができます。これは CLI の呼び出しで読み取られるデータ部分には影響しません。
- 個々のスタンザ・ヘッダー (キーワードの下) の後に、1 つのコロン (:) があります。例えば、`c175n05 resource info:` などです。
- 行の先頭の空白は特に意味がありません。タブやスペースは読みやすくするために使用されています。
- 最初の印刷可能文字としてポンド記号 (#) のある行はコメントです。
- 個々の行の各入力は空白 (タブまたはスペース) で区切られます。
- ファイル中のブランク行には特に意味はありませんが、読みやすくするために使用されています。
- 特定の `PersistentResourceAttributes` セクションに含めるリソース属性スタンザの数に制限はありません。
- 特定の `PersistentResourceClassAttributes` セクションに含めるリソース・クラス属性スタンザの数に制限はありません。通常、リソース・クラスにあるインスタンスは 1 つのみです。この場合、必要となるスタンザは 1 つです。
- 特定の `PersistentResourceAttributes` セクションにリソース属性スタンザが 1 つだけ含まれる場合は、`label:` 行を省略することができます。これは、`ResourceAction` セクションにも適用されます。
- 特定の `PersistentResourceClassAttributes` セクションにリソース・クラス属性スタンザが 1 つだけ含まれる場合は、`label:` 行を省略することができます。これは、`ResourceClassAction` セクションにも適用されます。
- スペースを含む値は引用符で囲む必要があります。
- 2 つのコロン (:) はセクションの終わりを示します。2 つの終了コロンが検出されない場合は、次の予約キーワードまたはファイルの終わりがセクションの終わりを通知します。
- 二重引用符で囲まれた文字列中で二重引用符を使用する場合は、エスケープ文字 `¥"` を使用する必要があります。

注: 一重引用符内では、二重引用符をネストできます。

例:

```
- "Name == ¥"testing¥"
- 'Name == "testing"'
```

その文字列が選択文字列であり、コマンド・ラインにカット・アンド・ペーストしたい場合は、この構文を使用します。

- 一重引用符で囲まれた文字列中で一重引用符を使用する場合は、エスケープ文字 `¥'` を使用する必要があります。

注: 二重引用符内では、一重引用符をネストできます。

以下に例を示します。

```
- 'Isn¥'t that true'
- "Isn't that true"
```

コマンド・ラインにカット・アンド・ペーストしたい場合は、この構文の使用が推奨されます。

- `resource_data_input` ファイルへのデータの入力に使用するフォーマットは、コマンド・ラインで使用するフォーマットと同じフォーマットにできないことがあります。コマンド実行先として選択したシェル

には、引用符に関する独自の規則があります。コマンド・ラインへのデータ入力の方法を決定するこれらの規則については、そのシェルの資料を参照してください。

実装上の固有な条件

この情報は AIX では `rsct.core.rmc` ファイルセットの一部であり、Linux、Solaris、および Windows では `rsct.core-3.1.0.0-0.platform.rpm` パッケージの一部です。ここで `platform` は、i386、ppc、ppc64、s390、または x86_64 です。

位置

`/opt/rsct/man/resource_data_input.7`

例

1. 次のサンプル `mkrsrc` コマンド

```
mkrsrc -f /tmp/my_resource_data_input_file IBM.Example
```

は、`IBM.Example` リソース・クラスのサンプル入力ファイル `/tmp/my_resource_data_input_file` を使用します。入力ファイルの内容は次のようになります。

```
PersistentResourceAttributes::
# Resource 1 - only set required attributes
resource 1:
    Name="c175n04"
    NodeList = {1}
# Resource 2 - setting both required and optional attributes
# mkrsrc -e2 IBM.Example displays required and optional
# persistent attributes
resource 2:
    Name="c175n05"
    NodeList = {1}
    Int32 = -99
    UInt32 = 99
    Int64 = -123456789123456789
    UInt64 = 123456789123456789
    Float32 = -9.89
    Float64 = 123456789.123456789
    String = "testing 123"
    Binary = 0xaabbccddeeff
    RH = "0x0000 0x0000 0x00000000 0x00000000 0x00000000 0x00000000"
    SD = [hello,1,{2,4,6,8}]
    Int32Array = {-4, -3, -2, -1, 0, 1, 2, 3, 4}
    Int64Array = {-4,-3,-2,-1,0,1,2,3,4}
    UInt32Array = {0,1,2,3,4,5,6}
    UInt64Array = {0,1,2,3,4,5,6}
    Float32Array = {-3.3, -2.2, -1.2, 0, 1, 2.2, 3.3}
    Float64Array = {-3.3, -2.2, -1.2, 0, 1, 2.2, 3.3}
    StringArray = {abc,"do re mi", 123}
    BinaryArray = {"0x01", "0x02", "0x0304"}
    RHArray = {"0x0000 0x0000 0x00000000 0x00000000 0x00000000 0x00000000",
               "0xaaaa 0xaaaa 0xbbbbbbbb 0xcccccccc 0xdddddddd 0xeeeeeeee"}
    SDArray = [[hello,1,{0,1,2,3}], [hello2,2,{2,4,6,8}]]
```

2. 次のサンプル `chrsrc` コマンド

```
chrsrc -f /tmp/Example/ch_resources -s 'Name == "c175n05"' IBM.Example
```

は、サンプル入力ファイル `/tmp/Example/ch_resources` を使用して、既存の `IBM.Example` リソースの属性値を変更します。入力ファイルの内容は次のようになります。

```
PersistentResourceAttributes::
# Changing resources that match the selection string entered
# when running chrsrc command.
resource 1:
    String          = "this is a string test"
    Int32Array      = {10,-20,30,-40,50,-60}
```

3. 次のサンプル **rmrsrc** コマンド

```
rmrsrc -l IBM.Examplebar
```

は、次のように、オプションのコマンド引数を表示します。

```
rmrsrc IBM.Examplebar ExampleInt32=int32 ExampleUint32=uint32
```

4. 次のサンプル **rmrsrc** コマンド

```
rmrsrc -f /tmp/Examplebar/rm_resources -s 'Name == "c175n05"' IBM.Examplebar
```

は、サンプル入力ファイル `/tmp/Examplebar/rm_resources` を使用して、**rmrsrc** コマンドのオプションのコマンド引数を指定します。入力ファイルの内容は次のようになります。

```
PersistentResourceArguments::
# Specifying command arguments when running rmrsrc command.
resource 1:
    ExampleInt32      = 1
    ExampleUint32    = 0
```

関連資料:

844 ページの『**rmrsrc** コマンド』

772 ページの『**rmccli** 情報ファイル』

関連情報:

chrsrc コマンド

lsactdef コマンド

mkrsrc 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、`/etc/security/ldap/ldap.cfg` ファイルに指定されている構成情報を **secdapclntd** デーモンが始動時に読み取ります。**secdapclntd** プロセスの開始時に、コマンド・ラインに以下のオプションが指定されている場合、コマンド・ラインのオプションは `/etc/security/ldap/ldap.cfg` ファイルの値を上書き

します。

項目	説明
-C <i>CacheSize</i>	secldapclntd 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・キャッシュの有効なエントリー数の範囲は 100 から 10,000 です。デフォルトは 1000 です。グループ・キャッシュのエントリー数は、ユーザー・キャッシュ・エントリー数の 10% です。
-o <i>LdapTimeout</i>	LDAP クライアントからサーバーへの要求のタイムアウト期間 (秒)。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の長さを決定します。有効範囲は 0 から 3600 (1 時間) です。デフォルトは 60 秒です。この値を 0 に設定すると、タイムアウトが使用不可になり、クライアントは無期限に待つことを強制されます。
-p <i>NumOfThread</i>	secldapclntd デーモンが使用するスレッド数を NumOfThread スレッドに設定します。有効範囲は 1 から 1000 です。デフォルトは 10 です。
-t <i>CacheTimeout</i>	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。
-T <i>HeartBeatIntv</i>	このクライアントと LDAP サーバーの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。

セキュリティ

aix.security.ldap 権限のあるユーザーが、このコマンドの使用を許可されます。

例

1. **secldapclntd** 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-secldapclntd
```

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、**secldapclntd** を再始動するには、次のように入力します。

```
/usr/sbin/restart-secldapclntd -p 30 -t 500
```

ファイル

項目	説明
/etc/security/ldap/ldap.cfg	secldapclntd デーモンがサーバーに接続する際に必要な情報が入っていません。

関連情報:

secldapclntd コマンド

mksecldap コマンド

stop-secldapclntd コマンド

start-secldapclntd コマンド

ls-secldapclntd コマンド

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

項目	説明
-o <i>File</i>	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 <i>Path</i>	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

項目	説明
/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。
/etc/objrepos/CuDv	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。
/etc/objrepos/CuAt	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。
/etc/objrepos/CuDep	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。
/etc/objrepos/CuDvDr	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

デバイス構成サブシステム: プログラミングの概要

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

デバイス構成コマンドのリスト

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

アーカイブされているファイルをファイル名別に復元するには、次のように入力します。

```
restore -x [ d M n O Q v q e ] [ -b Number ] [-L Label ] [-I Label ] [ -f Device ] [ -s SeekBackup ] [ -E { force | ignore | warn } ] [ File ... ]
```

アーカイブされているファイルをファイル名別にリストするには、次のように入力します。

```
restore -T | -t [ a l n q v Q ] [ -b Number ] [ -f Device ] [ -s SeekBackup ]
```

アーカイブされているファイルをファイルシステム別に復元するには、次のように入力します。

```
restore -r [ B O n q v y ] [ -b Number ] [ -f Device ] [ -s SeekBackup ]
```

アーカイブされているファイルをファイルシステム別に復元するには、次のように入力します。

```
restore -R [ B O n v y ] [ -b Number ] [ -f Device ] [ -s SeekBackup ]
```

アーカイブされているファイルをファイルシステム別に復元するには、次のように入力します。

```
restore -i [ O h m n q v y ] [ -b Number ] [ -f Device ] [ -s SeekBackup ]
```

アーカイブされているファイルをファイルシステム別に復元するには、次のように入力します。

```
restore -x [ B O 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を復元するには、次のように入力します。

```
restore -X Number [ -MdnqveOQ ] [ -b Number ] [ -f Device ] [ -s Number ] [ -E { force | ignore | warn } ] [ File ... ]
```

アーカイブされているファイルをファイルシステム別にリストするには、次のように入力します。

```
restore -t | -T [ B a l n h q v y ] [ -b Number ] [ -f Device ] [ -s SeekBackup ] [ File ... ]
```

アーカイブされているファイル属性をファイル名別に復元するには、次のように入力します。

```
restore -Pstring [ B d qv Q ] [ bNumber ] [ s SeekNumber ] [-L Label] [-I Label] [-f Device] [File ... ]
```

アーカイブされているファイル属性をファイルシステム別に復元するには、次のように入力します。

```
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、ディスクレット、テープに格納することができます。ファイルのアーカイブに使用されたのと同じ方法を使用してファイルを復元する必要があります。この操作を実行するには、アーカイブのフォーマットを知っている必要があります。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、または -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。

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。

ディスク・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/rfd0` のように、デバイスの範囲を指定することもできます。

注:

1. 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。
2. **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。
3. 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プ・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、アーカイブから抽出するファイルを指定するときにメタキャラクターを使用できるようになります。このプロセスでは、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抽出することができます。パターン・ファイル名は単一引用符で囲み、パターンは大括弧 [...] で囲む必要があります。

スパース・ファイルについて

オペレーティング・システムのファイルシステム内にある、長いヌル文字列を含むファイルは、他のファイルと比べて効率的に保管できます。ヌル文字列が割り当てブロック全体にまたがる場合、そのブロック全体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: 非スパース・ファイルを非スパースとして復元します。これらのファイルは、バック・ファイルとアンパック・ファイルの両方について、**backup** コマンドの名前フォーマットでアーカイブされたためです。ファイルをアーカイブする前に、復元される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。この確認が必要であるのは、**-e** フラグを使用可能にすると、このフラグはスパース・ファイルを非スパースとして復元するためです。このフラグを使用可能にする必要があるのは、復元されるファイルが 4 KB を超える大きさのヌルを含む非スパースである場合だけです。復元操作中に **-e** フラグが指定される場合、すべての通常ファイルが通常ファイルとして、また非スパース・データベース・ファイルが非スパースとして正常に復元されます。

フラグ

項目	ディスクリプター
-a	t および T 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ことを指定します。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。l (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	名前で行われるバックアップの 512 バイト・ブロックの数を指定します。i ノードで実行されるバックアップの場合、このフラグは、単一出力で読み取る 1024 バイト・ブロックの数を指定します。 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 <i>Number</i> 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 <i>Number</i> 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。 <i>Number</i> 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。 <i>Number</i> パラメーター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。
-d	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。
-e	<i>File</i> パラメーターがディレクトリーの場合、そのディレクトリー内のファイルをすべて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使用できます。 スパース・ファイルをアクティブに復元しないことを指定します。ファイルに、NULL が入っている調整済みでサイズ指定された区域のブロックがある場合、復元操作によって、それらのファイルシステム・ブロックが割り当てられて NULL で埋められるための物理スペースが作成されます。バイト単位で指定されるファイルのサイズは、ファイルシステム内で使用されるスペースに対応します。 このフラグは、4KB 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。

項目	ディスクリプター
-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ツ・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 エクステンツ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。
-f <i>Device</i>	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、警告を出します。入力デバイスを指定します。指定したデバイスから入力を受け取るには、 <i>Device</i> 変数をパス名 (/dev/rmt0 など) として指定します。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、 restore コマンドへの入力を dd コマンドからパイプ接続できます。
	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。
	<code>/dev/deviceXXX-YYY</code>
	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、8 mm 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。
	<i>Device</i> 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規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・フォーマットのときに使用できます。-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 <i>Label</i>	restore コマンドは、アーカイブにセキュリティー・ラベルのないファイルにこの保全性ラベルを適用します。適用されるラベルはシステム上に存在していなければなりません。このオプションは、Trusted AIX 上にある名前ファイルでファイルを復元する場合にのみ有効です。

項目
-i

ディスクリプター

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式に復元します。-i フラグ用のサブコマンドは、下記のとおりです。

cdDirectory

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。各ファイルがアーカイブから抽出される時、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。

-LLabel

restore コマンドは、アーカイブにセキュリティ・ラベルのないファイルにこの機密ラベルを適用します。適用されるラベルはシステム上に存在していなければなりません。このオプションは、Trusted AIX 上にある名前前でファイルを復元する場合にのみ有効です。

-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。この機能は、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。

項目	ディスクリプター
-n	<p>デフォルトでは、restore コマンドはアーカイブにある ACL、PCL、または名前付き拡張属性を復元します。-n フラグを指定すると、restore コマンドは、アーカイブ内の ACL、PCL、または名前付き拡張属性を無視し、復元しません。アーカイブ・ファイルに暗号化ファイルシステム (EFS) 情報が入っている場合は、-n フラグを指定しても EFS 拡張属性は復元されます。Trusted AIX システムでは、-n オプションを指定すると、restore コマンドは Trusted AIX セキュリティー属性を無視することになります。</p>
	<p>EFS の復元について詳しくは、「セキュリティ」のバックアップおよび復元を参照してください。</p>
-0	<p>restore コマンドは Trusted AIX セキュリティー属性を無視します。</p>
-Pstring	<p>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。</p>
	<p>A すべての属性を復元する。</p>
	<p>a ファイルの権限のみを復元する。</p>
	<p>o ファイルの所有権のみを復元する。</p>
	<p>t ファイルのタイム・スタンプのみを復元する。</p>
	<p>c ファイルの ACL 属性のみを復元する。</p>
	<p>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。</p>
	<p>注: 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。</p>
-Q	<p>名前によって実行されるバックアップの場合、エラーの発生時にコマンドが終了する必要があることを指定します。このプロセスでは、エラーが発生した場合には、リカバリーとアーカイブの処理の続行を試行しません。</p>
-q	<p>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」を表示し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。</p>
-r	<p>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、最終増分バックアップの復元時に削除する必要があります。-r フラグを使用すると、File パラメーターは無視されます。</p>
-R	<p>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。restore コマンドは、再開されると、-r フラグと同じように機能します。</p>
-sSeekBackup	<p>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、-s1 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的位置です。例えば、複数バックアップ・テープ・アーカイブから第 1、第 2、第 4 のバックアップを復元するには、-s フラグのそれぞれの値は -s1、および -s2 になります。</p>

項目 -t	<p>ディスクリプター</p> <p>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。 <i>File</i> 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。 <i>File</i> 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。 <i>File</i> 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。このフラグは、アーカイブがファイル名フォーマットであるか、またはファイルシステム・フォーマットであるかを判別するために使用できます。</p>
-T	<p>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、 <i>File</i> 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。</p>
-v	<p>ファイル名が復元されたときに情報を表示します。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。</p>

項目
-x

ディスクリプター

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ず。

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 コマンドにより、アーカイブから抽出するファイルを指定するときに使用されるシェル・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なります。マッチング・メタキャラクターに関する規則は、シェル・パス名「globbing」で使用される規則と同じです。すなわち、以下のとおりです。

* (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。

[] (大括弧)

大括弧で囲んだ文字のうちの任意の 1 文字と一致します。大括弧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、大括弧の中の「.」(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。

¥ (円記号)

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。

-XVolumeNumber

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。**restore** コマンドが開始されると、このコマンドは、-x フラグと同じように機能します。-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** コマンドは続行するための入力を要求します。どちらの場合も、読み取りバッファー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。

項目	ディスクリプター
0	正常終了。
>0	エラーが発生しました。

セキュリティ

Trusted AIX システムでは、権限のあるユーザーだけが **restore** コマンドを実行できます。

項目	ディスクリプター
aix.fs.manage.restore	このコマンドの実行に必要です。

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権命令を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権についての詳細情報は、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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適用されます。

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。

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る場合、アーカイブは、レベル 0 から始まって増加するバックアップ・レベル順 (0、1、および 2 など) に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、オープン時には緩み取りが行われず、クローズ時には巻き戻されるテープ・デバイスを指定します。-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 を指定する必要があります。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。

7. 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続いており、ファイルが復元されていない場合、**restore** コマンドは、バックアップの終了点に達するまで次のボリュームを挿入するよう指示します。-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、次の **backup** コマンドは、この例で取り出されるアーカイブを作成するためには使用できません。

```
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、次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

項目	ディスクリプター
/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

mkfs コマンド

fsck コマンド

rmt コマンド

System Management Interface Tool

Trusted AIX

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

項目	説明
-b blocks	blocks 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。 blocks 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。
-f device	device 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) を取めているデバイスのタイプを指定します。 -f を指定していない場合は、 device はデフォルトの /dev/rmt0 になります。
-a	-b block 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。 -a 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。
-n	ACL、PCL、あるいは拡張属性を、復元しません。
-s	バックアップ・ソースがユーザー・ボリューム・グループで、 rootvg でないことを指定します。
-d path	path 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。 -d 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 root の場合は、問題が起きます。 root 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。
-D	デバッグの出力を生成します。

パラメーター

項目	説明
file_list	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、 root 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
```

4. `/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

項目	説明
<code>/usr/bin/restorevgfiles</code>	<code>restorevgfiles</code> 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ってはなりません。

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

項目	説明
-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。
DiskName...	vgname.data 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、 mkdev 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、 reducevg 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。
-d FileName	-d 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 vgname.data 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。
-f Device	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 /dev/rmt0 です。
-l	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 -f device フラグを必要とします。このフラグを指定すると、 restvg 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 uname 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。 -l 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 " lsvg -l vgname " を実行したものに相当します。
-n	既存の MAP ファイルが無視されることを指定します。 -n フラグは、 vgname.data ファイルの logical_volume_policy スタンザ内の EXACT_FIT フィールド値を指定変更します。
-P PPSize	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、 restvg は、復元先の 1 番大きいディスクに合わせて PPsize にその最大値を使用します。この値が vgname.data 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 PPsize 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 PPsize が指定されている場合は、大きい方の PPsize が使用されます。 適切なディスク・サイズよりも大きい PPsize が指定されている場合は、指定の大きい方の PPsize が使用されます。
-q	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す。
-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。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、すべての AIX 論理ボリューム構造を決まった位置で必要とするユーザーにとっては、このフラグは便利です。 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。

項目

-s

説明

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
```

注: ボリューム・グループのバックアップについて詳しくは、`listvgbackup` コマンドを参照してください。個々のファイルをボリューム・グループ・バックアップからリストアするには、`restorevgfiles` コマンドを参照してください。

関連資料:

726 ページの『`restore` コマンド』

683 ページの『`reducevg` コマンド』

関連情報:

`mkvgdata` コマンド

`savevg` コマンド

restwpar コマンド

目的

workload partitionを復元します。

構文

```
restwpar [ -a ] [ -A ] [ -b Blocks ] [ -B devexportsFile ] [ -C ] [ -d Directory ] [ -f Device ] [ -F ] [ -h
hostName ] [ -i imagedataFileName ] [ -k ] [-K] [-M mkwparFlags] [ -n WparName [ -r ] [ -s ] [ -S { a
| A | f | F | n } ] [ -U ] [ -w wparSpecificationFile ]
```

説明

restwpar コマンドは、**savewpar**、**mkcd**、または **mkdvd** コマンドで作成されたworkload partitionバックアップ・イメージからworkload partitionを作成します。

警告: **restwpar** コマンドは、AIX Live Update 操作の進行中に実行しないでください。

workload partitionバックアップ・イメージには、workload partition *WparName* の特性を設定するために使用される *image.data* ファイルおよびworkload partition仕様ファイルが入っています。これらのデフォルト・ファイルは、**-i** フラグおよび **-w** フラグを使用してオーバーライドできます。

-f フラグを指定しない場合、入力デバイスとして */dev/rmt0* デバイスが使用されます。

/tmp/wpardata/WparName/image.data ファイルにある *logical_volume_policy* スタンザの *EXACT_FIT* フィールドに値 *Yes* を指定した場合、**restwpar** コマンドは、マップ・ファイルを使用して各論理ボリュームの物理区画の配置を保持します。

注: バックアップ・イメージのファイルを表示したり、バックアップ・イメージから個別のファイルを復元したりするには、**-T** フラグまたは **-x** フラグを指定した **lssavewpar**、**restwparfiles**、または **restore** コマンドを使用してください。

フラグ

項目	説明
-a	必要なら、静的設定の競合を自動的に解決します。解決可能な設定は、名前、ホスト名、ベース・ディレクトリー、およびネットワーク構成です。
-A	システムの起動のたびに実行されるようにグローバル <i>/etc/inittab</i> に追加された <i>/etc/rc.wpars</i> コマンドが実行されるたびに、workload partitionを開始します。デフォルトでは、workload partitionは自動的に開始されません。
-bBlocks	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> パラメーターを指定しない場合、 restore コマンドはデフォルト値 100 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。
-BdevexportsFile	マスター・デバイス・エクスポート・ファイルとして使用できる代替ファイルを指定します。このファイルは、デバイス・エクスポート・ファイルのフォーマットに一致しなければなりません。ファイル名を指定しない場合は、 <i>/etc/wpars/devexports</i> ファイルが使用されます。

項目	説明
-C	バックアップ・イメージのシステムとバックアップが復元されるシステム間の互換性検査が失敗した場合にも、指定されたworkload partitionの作成を強制します。
	ターゲット・システムと互換性がないworkload partitionは、操作できない場合があります。
	グローバル・システムのオペレーティング・システムが、VRMF (バージョン、リリース、モディフィケーション、およびフィックスのレベル) のモディフィケーションまたはフィックスのレベルが異なる WPAR より後のテクノロジー・レベルまたは Service Pack レベルである場合、workload partition (WPAR) を新しいグローバル・システムと同期できます。同期の正常実行には、さまざまな要因が影響します。同期操作が完了した後、ログを確認してください。新しいグローバル・システムに適用されたすべての更新をコミットする必要があり、WPAR をバックアップする前に WPAR に対する更新をコミットする必要があります。AIX 6100-08 または 7100-02 以前のテクノロジー・レベルを稼働しているシステムに新しいグローバル・システムがインストールされる場合、同期が機能するためには、workload partitionを復元する前に cp_bos_updates コマンドを実行する必要があります。
-d <i>Directory</i>	workload partitionのベース・ディレクトリーを指定します。ディレクトリー名を指定しない場合は、WPAR 仕様ファイルにあるディレクトリー名が使用されます。
-f <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。デフォルト値は /dev/rmt0 です。
-F	指定されたworkload partitionの作成を強制します。指定されたworkload partitionが存在する場合、新しいworkload partitionが作成される前に、既存の区画が (アクティブなら停止してから) 除去されます。
-hostname	workload partitionのホスト名を指定します。指定しない場合、 mkwpar コマンドはworkload partition名をホスト名として使用します。
-i <i>imagedataFileName</i>	ファイル名を指定するオプションのフラグ。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 image.data ファイルの代わりとして使用されます。
-k	最小サイズの論理ボリュームをバックアップから作成します。
-K	インストール後のカスタマイズ・スクリプトを作成します。
-M <i>mkwparFlags</i>	workload partitionを作成するために直接 mkwpar コマンドに渡されるフラグを指定します。他のフラグを mkwpar コマンドに渡すためには、 -M フラグを使用します。フラグがそれ自身のオプションおよび -M フラグを通して渡される場合、両方のフラグが mkwpar コマンドに渡されます。
	注: <i>mkwparFlags</i> 値に -i フラグと -f フラグを含めることはできません。これらのフラグは restwpar コマンド用に予約済みです。 <i>mkwparFlags</i> の値として -i フラグまたは -f フラグを指定するとエラーになります。
-n <i>WparName</i>	作成するworkload partitionの名前を指定します。 -n フラグを指定しない場合、 <i>WparName</i> は WPAR 仕様ファイルから取られます。
-r	グローバル・システムからネットワーク・ネーム・レゾリューション構成を複写します。以下のファイルが存在する場合は、workload partitionにコピーされます。
	<ul style="list-style-type: none"> • /etc/resolv.conf • /etc/hosts • /etc/netsh.conf • /etc/irs.conf • /etc/networks
	呼び出し環境で NSORDER 環境変数が定義されている場合、workload partitionの /etc/environment ファイルにこの環境変数が追加されます。
-s	作成されたworkload partitionを開始します。

項目	説明
-S { a A f F n }	<p>ファイルがバックアップから復元された後で、workload partition内のソフトウェアのレベルとグローバル環境内のソフトウェアのレベルを同期化するために使用される同期のタイプを指定します。</p> <p>a ソフトウェアを除去せずに追加のインストールを行います。このオプションはデフォルトです。</p> <p>A ソフトウェアを除去せずに追加のインストールを行い、同期エラーはすべて無視します。 重要: -S A を指定すると、workload partitionが使用できない状態になる場合があります。</p> <p>f 追加のインストール、ソフトウェアの除去、およびアンインストールを行います。</p> <p>F 追加のインストール、ソフトウェアの除去、およびアンインストールを行います。このオプションは、同期のエラーをすべて無視します。 重要: -S F を指定すると、workload partitionが使用できない状態になる場合があります。</p> <p>n ファイルが復元された後の同期化処理を防ぎます。 重要: -S n を指定すると、workload partitionが使用できない状態になる場合があります。</p>
-U	既存の MAP ファイルが無視されることを指定します。-U フラグは、WparName.data ファイルの logical_volume_policy スタンザ内の EXACT_FIT フィールドの値をオーバーライドします。
-w wparSpecificationFile	ファイル名を指定するオプションのフラグ。このファイルは、mkwpar コマンドにより、WPAR 仕様ファイルとして WPAR バックアップ・イメージ内のバージョンの代わりに使用されます。

例

- workload partitionイメージを /dev/rmt1 デバイスから復元するには、次のコマンドを入力します。
restwpar -f/dev/rmt1
- /mydata/wpar.img ファイルに保管されているworkload partitionイメージを mywpar という名前でベース・ディレクトリー /wpars/mywpar に復元するには、次のコマンドを入力します。
restwpar -f/mydata/wpar.img -n mywpar -d /wpars/mywpar
- workload partitionイメージを /dev/usbms0 デバイスから復元するには、次のコマンドを入力します。
restwpar -f/dev/usbms0

関連情報:

mkwpardata コマンド

restore コマンド

savewpar コマンド

mkwpar コマンド

restwparfiles コマンド

目的

workload partitionバックアップ・ソースからファイルを復元します。

構文

```
restwparfiles [ -b blocks ] [ -f device ] [ -a ] [ -m ] [ -n ] [ -d path ] [ -D ] [-V] [ file_list ]
```

説明

restwparfiles コマンドは、テープ、ファイル、CD-ROM、またはその他のworkload partitionバックアップ・ソースからファイルを復元します。**restwparfiles** 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。

フラグ

項目	説明
-a	-b blocks フラグで指定されたテープ・バックアップの物理ブロック・サイズを検査します。バックアップを読み取るためにブロック・サイズの変更が必要になる場合があります。 -a フラグは、 -f フラグでデバイスとしてテープを指定した場合にのみ有効です。
-b blocks	blocks 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。 blocks パラメーターを指定しない場合は、読み取るブロック数はデフォルトの 100 になります。
-d path	ファイルの復元先のディレクトリー・パスを、 path パラメーターで定義して指定します。 -d フラグを指定しない場合は、現在の作業ディレクトリーが使用されます。 制限: ファイルの復元先のディレクトリー・パスは、グローバル環境のルート (<i>/</i>) であってはなりません。つまり、 -d / を指定することはできません。また、現在の作業ディレクトリーが <i>/</i> の場合に -d フラグを指定しないこともできません。
-D	デバッグの出力を生成します。
-f device	バックアップが入っているデバイス (ファイル、テープ、CD-ROM、またはその他のソース) を、 device パラメーターで定義して指定します。 -f フラグを指定しない場合、デフォルト・デバイスは /dev/rmt0 です。
-m	情報ファイルと制御ファイルのみをイメージから復元します。バックアップ・イメージから image.data ファイルと wpar.spec ファイルを復元するには、このフラグを使用します。ファイルは ./savewpar_dir/ ディレクトリーの下に復元されます。
-n	ACL、PCL、または拡張属性を復元しないことを指定します。
-V	テープ・バックアップを検査します。 -V フラグには -f device フラグの指定が必要であり、テープ・デバイスの指定にのみ使用できます。 -V フラグを指定すると、 restwparfiles コマンドは、ボリューム・グループ・バックアップの各ファイル・ヘッダーの読み取り可能性を検査し、発生したエラーはすべて標準エラー・ログ (stderr) に出力されます。

パラメーター

項目	説明
file_list	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を、スペースで区切ったリストとして指定します。リストを省略すると、指定されたディレクトリー内のすべてのファイルが復元されます。1 つのディレクトリー内のすべてのファイルを復元する場合、ルート・ディレクトリーではなく一時フォルダーに書き込んでください。

例

1. **/dev/cd1** デバイス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。

```
restwparfiles -f /dev/cd1 -d /data/myfiles
```

2. デフォルト・デバイスから一度に 20 個の 512 バイト・ブロックのバックアップを読み取り、**/myapp/app.c** ファイルを現行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。

```
restwparfiles -b 20 ./myapp/app.h
```

3. **/dev/cd1** デバイスに保存されている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、次のコマンドを入力します。

```
restwparfiles -f /dev/cd1 -d /data/testcode ./myapp/app.c
```

4. `/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` ディレクトリに復元するには、次のコマンドを入力します。

```
restwparfiles -f /dev/usbms0 -d /data/myfiles
```

関連情報:

lssavewpar 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・グループに対して定義された 1 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。**-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 **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
```

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

項目	説明
<code>/usr/bin/rev</code>	<code>rev</code> コマンドが入っています。

関連情報:

入出力リダイレクト

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

項目	説明
<code>-h</code>	netgroup.byhost マップを作成するための出力を生成します。
<code>-u</code>	netgroup.byuser マップを作成するための出力を生成します。

例

1. `/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
```

2. `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

項目	説明
<code>/etc/netgroup</code>	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。
<code>/var/yp/Makefile</code>	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・ファイルシステム (NFS) の概要

NIS のリファレンス

rexid 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

rexid 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 **rexid** 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。**rexid** 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 **rexid** 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

項目	説明
<code>/tmp_rex/rexd</code>	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。
<code>/etc/exports</code>	サーバーがエクスポートできるディレクトリーのリストが入っています。
<code>inetd.conf</code>	RPC デーモンおよびその他の TCP/IP デーモンを始動します。
<code>/etc/passwd</code>	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。

関連資料:

757 ページの『`rlogin` コマンド』

関連情報:

`inetd` コマンド

ネットワーク・ファイルシステム (NFS) の概要

NFS コマンドのリスト

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 (`securetcpip` 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・ラインに `-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。

制限: ユーザー ID が 128 以下のユーザーは、リモート Trusted AIX システムにログインできません。

フラグ

項目	説明
<code>-a</code>	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。
<code>-d</code>	ソケット・レベルのデバッグを使用可能にします。
<code>-i</code>	<code>stdin</code> を読み取れないようにします。
<code>-n</code>	自動ログインできないようにします。 <code>-n</code> フラグを指定すると、 <code>rexec</code> コマンドは <code>\$HOME/.netrc</code> 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

項目	説明
コマンド	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。

関連資料:

757 ページの『**rlogin** コマンド』

『**rexecd** デーモン』

関連情報:

通信およびネットワーク

securetcip コマンド

.netrc コマンド

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: 通常、**rexecd** デーモンは、**/etc/inetd.conf** コマンドまたは **kill -1 InetdPID** コマンドによって起動され、**inetd** デーモンにその構成ファイルに対する変更内容を通知します。

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

項目	説明
-s	ソケット・レベルのデバッグを使用可能にします。
-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。

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

startsrc コマンド

inetd コマンド

/etc/inetd.conf コマンド

TCP/IP プロトコル

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作成します。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

項目	説明
<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

項目	説明
<i>/usr/lib/X11/rgb.txt</i>	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

項目	説明
-1	照会をバージョン 1 パケットとして送信します。
-2	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)。
-[a5] authkey	照会に使用する認証パスワードを指定します。 -a を指定すると認証タイプ SIMPLE が使用され、 -5 を指定すると認証タイプ MD5 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 NONE 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。
-n	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。
-N dest[/mask]	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。
-p	経路指定テーブルの情報を要求するのに RIP POLL コマンドを使用します。これはデフォルトです。 RIP POLL コマンドへの応答がないと、 RIP REQUEST コマンドが試行されます。 gated は、RIP を介して得たすべての経路を表示して POLL コマンドに応答します。
-r	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 REQUEST コマンドを使用します。 RIP POLL コマンドとは異なり、すべてのゲートウェイ で RIP REQUEST をサポートする必要があります。 RIP REQUEST コマンドへの応答がないと、 RIP POLL コマンドが試行されます。 gated 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 REQUEST コマンドに応答します。
-v	ripquery 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。
-w time	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 コマンド

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、左脱字記号)、または >> (右脱字記号が 2 つ) が含まれるコマンドの出力をリダイレクトすること。

フラグ

項目	説明
-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、 <i>String</i> 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 <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。
-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。

項目
-o Option

説明
現在のオプション設定、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。

erexit

-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。

モニター

-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・ラインに複数のオプションを設定することができます。

-p

ログイン・シェルとしてシェルを使用時に、**\$HOME/profile** ファイルの処理を使用不可にします。

-s

rksh 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、-c 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。

-t

1 つのコマンドを読み込み、実行後、終了します。

-u

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。

-v

シェル入力行を、読み込まれたとおりに印刷します。

-x

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

項目	説明
<code>/usr/bin/rksh</code>	制限付き Korn シェルへのパス名が入っています。
<code>/tmp/sh*</code>	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:

ksh コマンド
通信およびネットワーク
制限付き Korn シェル
shells コマンド

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、指定されたリモート・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・ライン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** and **/etc/security/login.cfg** ファイルで行います。

POSIX 伝送制御手順

rlogind デーモンおよび **telnetd** デーモンは、POSIX 伝送制御手順を使用して、ローカル TTY 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

項目	説明
-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。このような目的では、レルムは DCE セルと同義です。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	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・ユーザー・アカウントにある、リモートの **\$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。html を参照してください。

関連資料:

648 ページの『rcp コマンド』

750 ページの『rexec コマンド』

関連情報:

ftp コマンド

.rhosts コマンド

認証とセキュア rcmds

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・ライン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 デーモンの変更は、System Management Interface Tool (SMIT) または System Resource Controller (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・ラインに **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。

項目	説明
Try again.	サーバーが fork コマンドを実行しましたが失敗しました。
/usr/bin/shell:	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

項目	説明
-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 **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 -f -t 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

startsrc コマンド

/etc/inetd.conf コマンド

TCP/IP デーモン

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

項目	説明
-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せん。

項目	説明
-r	<i>File</i> 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	<i>File</i> 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。

項目	説明
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ト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トリーがすべて除去されます。
>0	エラーが発生しました。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
<code>/usr/bin/rm</code>	rm コマンドが入っています。

関連情報:

`ln` コマンド

`unlink` コマンド

ディレクトリー・コマンド

ファイルとディレクトリーのアクセス・モード

ファイルおよびディレクトリー・リンク

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

メールの管理

rmaild コマンド

目的

1 つ以上のユーザー定義の権限を除去します。

構文

rmaild [-**R** *load_module*] [-**h**] *Name*

説明

rmauth コマンドは、*Name* パラメーターで指定されたユーザー定義の権限を除去します。このコマンドは、権限データベースに存在するユーザー定義の権限のみを除去します。このコマンドでシステム定義の権限を除去することはできません。権限が特権コマンド・データベース内で参照中の場合、データベースによる参照が終わるまでその権限は除去できません。

デフォルトでは、**rmauth** コマンドは指定された権限を権限データベースから除去することを試みるだけです。階層内で高いレベルにある権限を除去するには、まず一番低いレベルから権限を除去する必要があります。低レベルの権限がまだ存在する場合に高レベルの権限の除去を指定すると、コマンドは失敗します。権限の階層を除去するには、**-h** フラグを指定します。**-h** フラグを指定すると、指定された権限より低レベルの権限もすべて除去されます。低レベルの権限が特権コマンド・データベース内で参照されている場合、権限は 1 つも除去されず、操作全体が失敗します。

複数のドメインからデータベースを使用するようにシステムが構成されている場合、**rmauth** コマンドは、*/etc/nscontrol.conf* ファイル内の対応する権限スタanzasの **secorder** 属性で指定された順序で、データベース・ドメインから最初の一致を検索します。それと同時に、**rmauth** コマンドは、検索された権限のエントリをドメインから除去します。残りのドメインに一致する権限があっても、それらの権限は影響されません。特定のドメインから権限を除去するには、**-R** フラグを使用します。

システムが拡張された役割ベースのアクセス制御 (RBAC) モードで作動している場合、権限データベースに対して行われた変更は、**setkst** コマンドを使用してデータベースがカーネル・セキュリティ・テーブルに送られるまで、セキュリティ上の考慮には使用されません。

フラグ

項目	説明
-h	権限の階層の除去を許可します。
-R load_module	権限の削除に使用するロード可能モジュールを指定します。

パラメーター

項目	説明
名前	除去する権限を指定します。

セキュリティ

rmauth コマンドは特権コマンドです。このコマンドを実行するためには、**aix.security.role.remove** 権限が必要です。

項目	説明
aix.security.auth.remove	このコマンドの実行に必要です。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

アクセスされるファイル

ファイル	モード
/etc/security/authorizations	rw

例

1. `custom.test` 権限を除去するには、次のコマンドを入力します。
`rmauth custom.test`
2. `custom` 権限とその子権限をすべて除去するには、次のコマンドを入力します。
`rmauth -h custom`
3. LDAP から `custom.test` 権限を除去するには、次のコマンドを入力します。
`rmauth -h custom.test`

関連情報:

`mkauth` コマンド

`putauthattrs` コマンド

`/etc/security/authorizations` コマンド

`/etc/nscontrol.conf` コマンド

RBAC コマンド

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、システムのオペ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、「Administering RSCT」ガイド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。

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。

#mmddhhmmyyyy

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。

#-mmddhhmmyyyy

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。

-Ssubsystem_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・ライン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・ライン・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・ラインに入力したフラグが誤りです。
- 4 コマンド・ラインに入力したパラメーターが誤りです。
- 5 誤ったコマンド・ライン入力により、エラーが起きました。

環境変数

CT_CONTACT

Resource Monitoring and Control (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

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 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

`/opt/rsct/bin/rmaudrec`

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、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

項目	説明
<code>-m</code>	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・ライン・オプションが使用されました。

ファイル

項目	説明
<code>/usr/sbin/rmCC2admin</code>	<code>rmCC2admin</code> コマンドが入っています。

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

項目	説明
<code>-m</code>	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・ライン・オプションが使用されました。

ファイル

項目	説明
<code>/usr/sbin/rmCCAdmin</code>	<code>rmCCAdmin</code> コマンドが入っています。

rmccli 情報ファイル

目的

Resource Monitoring and Control (RMC) と関連するコマンドについての一般情報を提供します。

説明

RMC と関連するコマンドについての一般情報。この情報にはデータ型や用語が含まれ、さらに、関連情報へのリファレンスが続きます。

コマンドの構造および使用

RMC コマンドは、リソース・クラスおよびリソースで実行できる、さまざまな操作を表すカテゴリにグループ化されます。

- リソースの作成と除去: `mkrsrc`、`rmrsrc`
- リソースの変更: `chsrc`、`refsrc`
- 定義およびデータの表示: `lsrsrc`、`lsrsrcdef`
- アクションの表示: `lsactdef`
- アクションの実行: `runact`

RMC コマンドはコマンド・ラインから直接実行することも、ユーザー作成スクリプトによって呼び出すこともできます。さらに、RMC コマンドは、イベント応答リソース・マネージャー (ERRM) コマンドのような高水準コマンドの基本としても使用されます。

データ表示情報

RMC CLI ルーチンの表示機能を優先順位に従って制御するフラグは次のとおりです。

1. 長形式の表示の場合は `-l` です。このフラグは、デフォルトの表示形式です。

例えば、次のコマンド

```
lsrsrc -s 'Name == "c175n05"' IBM.Foo Name NodeList SD Binary RH Int32Array
```

を出した場合、以下の出力が作成されます。

```
Persistent Attributes for Resource: IBM.Foo
resource 1:
```

```
Name           = "c175n05"
NodeList        = {1}
SD              = ["testing 1 2 3",1,{0,1,2}]
Binary          = "0xaabbcc00 0xeeff"
RH              = "0x0000 0x0000 0x00000000 0x00000000 0x00000000 0x00000000"
Int32Array      = {1,5,-10,1000000}
```

2. 表形式の表示の場合は `-t` です。

例えば、次のコマンド

```
lsrsrc -s 'Name ?= "Page"' -t IBM.Condition Name EventExpression
```

を出した場合、以下の出力が作成されます。

```
Persistent Attributes for Resource: IBM.Condition
```

Name	EventExpression
"Page space out rate"	"VMPgSpOutRate > 500"
"Page fault rate"	"VMPgFaultRate > 500"
"Page out rate"	"VMPgOutRate > 500"
"Page in rate"	"VMPgInRate > 500"
"Page space in rate"	"VMPgSpInRate > 500"

3. 印刷時にヘッダーを抑止する場合は `-x` です。

4. コロン (:) 区切り表示の場合は `-d` です。

例えば、次のコマンド

```
lsrsrc -xd -s 'Name == "c175n05"' IBM.Foo Name Int32 Uint32Array SD Binary
```

を出した場合、以下の出力が作成されます。

```
c175n05:-100:{:}:[ "hel lo1",1,{0,1,2}]:"0xaabbcc00 0xeeff":
```

`-x` フラグと一緒に `-d` フラグを使用することに注意してください。

5. 文字列区切り表示の場合は `-Ddelimiter` です。

例えば、次のコマンド

```
lsrsrc -xD:: -s 'Name == "c175n05"' IBM.Foo Name Int32 Uint32Array SD Binary
```

を出した場合、以下の出力が作成されます。

```
c175n05:::-100::{:}::[ "hel lo1",1,{0,1,2}>:::"0xaabbcc00 0xeeff"::
```

`-x` フラグと一緒に `-DDelimiter` を使用することに注意してください。

リスト・コマンド `lsrsrc lsrsrcdef` の出力が表形式の出力フォーマットで表示される場合、印刷列の幅が切り捨てられることがあります。表示する必要がある文字がさらにある場合 (文字列の場合) には、そのフィールド全体を表示するために `-l` フラグを使用します。

データ入力フォーマット

バイナリー形式の属性用バイナリー・データは、次のフォーマットで入力できます。

- "0xxxxxxxxxxxx 0xxxxxxxxxxxx 0xxxxxx..."
- "0xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx..."
- 0xxxxxxxxxxxxxxxxxxxxxxxxxxxxx...

いずれかの整数データ型の属性用の整数データは、次のフォーマットで入力できます。

- ゼロ以外の数字で始まる 10 進定数 (例えば Int32=45)
- プレフィックス 0 で始まる 8 進定数。これには 0 から 7 の範囲の 10 進数の組み合わせが続く場合があります (例えば Int32=055)
- プレフィックス 0x または 0X で始まる 16 進定数と、後続の a から f および A から F の範囲の 10 進数の組み合わせ (例えば Int32=0x2d)

文字列を入力データとして指定するときには、十分に注意してください。文字列は、含まれる文字によって以下のような制約があります。

- 空白以外の文字または非英数字は、引用符で囲まずに入力として入力することができます。
- 空白または他の英数字は引用符で囲む必要があります。
- 一重引用符 (') は、次の例に示すように二重引用符 (") で囲む必要があります。"this is a string with 'single quotation marks'"

選択文字列内に二重引用符がない限り、選択文字列を二重引用符で囲む必要があります。二重引用符がある場合、選択文字列を一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、「Administering RSCT」ガイドを参照してください。

- 選択文字列の入力例は次のようになります。"NodeNumber == 1"
- 二重引用符が選択文字列の一部である選択文字列の入力は次のようになります。'Name == "c175n05"'

構造化データ (SD) 型は、大括弧で囲む必要があります。[hello,1,{2,4,6,8}]

構造化データ (SD) を RMC コマンドへのコマンド・ライン入力として指定するときには、その SD を次のように一重引用符で囲む必要があります。SD='[hello,1,{2,4,6,8}]'

すべてのタイプの配列は中括弧 {} で囲む必要があります。

- 整数の配列: {-4, -3, -2, -1, 0, 1, 2, 3, 4}
- 文字列の配列: {abc, "do re mi", 123}
- 構造化データの配列: {[hello,1,{0,1,2,3}], [hello2,2,{2,4,6,8}]}

複数のエレメントを持つすべてのタイプの配列は、引用符で囲む必要があります。以下に例を示します。

- **mkrsrc** IBM.Foo Name=testing NodeList={1} Uint32Array='{1,2,3}'
- **mkrsrc** IBM.Foo Name=testing NodeList='{1}' Uint32_array='{1,2,3}'

文字列の配列および構造化データの配列は、常に引用符で囲む必要があります。

構造化データの配列、または引用符で囲まれた文字列を含む配列を RMC コマンドへのコマンド・ライン入力として指定するときには、その配列全体を一重引用符で囲みます。

- 文字列の配列: mkrsrc IBM.Foo Name="c175n05" NodeList={1} StringArray='{"a string","a different string"}'
- 構造化データの配列: mkrsrc IBM.Foo Name="c175n05" NodeList={1} SDArray='{["string 1",1,{1,1}],["string 2",2,{1,2,3}]}'

詳細については、resource_data_input を参照してください。

データ出力フォーマット

文字列データは常に、次のように二重引用符または一重引用符のいずれかで囲まれて表示されます。

- 文字列「This is a string that contains white space」と等しい記述属性は、次のように長形式で表示されます。

Description = "This is a string that contains white space"

- 空の文字列 "" と等しい記述属性値は次のように長形式で表示されます。

Description = ""

- 文字列の終わりに改行文字を含む文字列と等しい記述属性値は次のように長形式で表示されます。

Description = "This string ends with a new-line character..."

- 二重引用符を含む選択文字列は次のように長形式で表示されます。

SelectionString = 'Name == "c175n05"'

- 文字列「c175n05」と等しい名前属性値は次のように長形式で表示されます。

Name = "c175n05"

バイナリー・データは次のように表示されます。

"0x nnnnnnnn 0x nnnnnnnn 0x nnnnnnnn 0x nnnnnnnn"

命名規則

次の変数名は RMC コマンド・マニュアル・ページ全体で使用されます。

変数	説明
<i>attr</i>	リソース・クラスまたはリソース属性の名前
<i>resource_class</i>	リソース・クラスの名前

ノード・グループ

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。

用語

attribute

属性は、永続的または動的のいずれかです。リソース・クラスは、永続属性と動的属性のセットによって定義されます。リソースも、永続属性と動的属性のセットによって定義されます。永続属性は、リソース・クラスおよびリソースの構成を定義します。動的属性は、リソース・クラスおよびリソースの状態またはパフォーマンス関連の側面を定義します。同じリソース・クラスまたはリソース内では、属性名を永続的または動的のいずれかとして指定可能であるが、両方は指定できません。

resource

システム内でのサービスのセットを提供するエンティティ。ハードウェア・エンティティには、プロセッサ、ディスク・ドライブ、メモリー、アダプターがあります。ソフトウェア・エンティティには、データベース・アプリケーション、プロセス、ファイルシステムがあります。システム内の各リソースには、リソースの状態を定義する 1 つ以上の属性が存在します。

リソース・クラス

ノード、ファイルシステム、アダプターなどのシステム・リソースの広範なカテゴリー。各リソース・クラスにはそのリソース・クラスに適用される機能、情報、動的属性、および条件を取めるコンテナーがあります。例えば、『/tmp space used』条件はファイルシステムのリソース・クラスに適用されます。

リソース・マネージャー

リソースおよびリソース・クラス抽象を 1 つ以上の特定のタイプのリソースの呼び出しおよびコマンドにマップするプロセス。リソース・マネージャーはスタンドアロン・デーモンになることも、アプリケーションまたはサブシステムに直接統合することも可能です。

システムに定義されているすべてのリソース・クラスを表示するには、フラグまたはパラメーターを指定せずに **lsrsrc** コマンドを実行します。IBM.FileSystem リソース・クラスのシステムに定義されているすべてのリソースを表示するには、以下のように入力します。

```
lsrsrc IBM.FileSystem
```

選択文字列

二重引用符または一重引用符で囲む必要があります。選択文字列に二重引用符が含まれている場合は、その選択文字列全体を一重引用符で囲んでください。例えば、次のようになります。

```
-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。

フラグ

- h** コマンドの使用法のステートメントを標準出力に書き込みます。
- T** 標準エラーにコマンドのトレース・メッセージを書き込みます。これはソフトウェア・サービス組織専用です。
- V** コマンドの詳細メッセージ (使用可能なメッセージがある場合) を標準出力に書き込みます。

すべての RMC コマンドには、**-T** フラグと **-V** フラグが組み込まれます。**-T** フラグを使用するのは、ソフトウェア・サービス組織からトレースをオンにするよう指示された場合のみです。トレース・メッセージは変換されません。コマンドの詳細を表示するには、「詳細」モードを示す **-V** フラグを使用します。詳細メッセージ (使用可能なメッセージがある場合) は、メッセージ・カタログに入っており、実行している地域またはその他の基準に従って変換されます。

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 (Resource Monitoring and Control) デーモンに連絡します。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。コマンドによって表示または変更されるリソース・クラスやリソースは、接続が確立されたシステム上にあります。

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 環境変数は、CT_CONTACT 環境変数が IP アドレスに設定されている場合に有効です。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。

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 フラグが指定されている場合は、これらのコマンドの詳細メッセージ (使用可能なメッセージがある場合) 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。

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・ライン・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・ラインに間違っただフラグが指定されました。
- 4 コマンド・ライン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・ライン入力に基づく RMC により、エラーが発生しました。
- 6 指定された選択文字列に一致するリソースが見つかりませんでした。

セキュリティ

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。

実装上の固有な条件

この情報は AIX では *rsct.core.rmc* ファイルセットの一部であり、Linux、Solaris、および Windows では *rsct.core-3.1.0.0-0.platform.rpm* パッケージの一部です。ここで *platform* は、i386、ppc、ppc64、s390、または x86_64 です。

位置

`/opt/rsct/man/rmccli`

`/opt/rsct/man/rmccli.7` (Linux プラットフォームの場合)

rmcctrl コマンド

目的

Resource Monitoring and Control (RMC) サブシステムを管理します。

構文

```
rmcctrl { -a | -A | -b | -B | -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、Resource Monitoring and Control (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、システムの始動時に RMC サブシステムが自動的に始動されるように、**/etc/inittab** ファイルにエントリーが作成されます。

注: RMC サブシステムの停止および始動は **stopsrc** コマンドおよび **startsrc** コマンドで行えますが、**rmcctrl** コマンドでは、以下の機能を実行できます。

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。
- b** RMC API クライアント・セッションのアイドル・タイムアウトを **n** 秒に設定します。RMC デーモンが最後の **n** 秒間にセッションでアクティビティを検出しない場合は、クローズされます。
- B** RMC API クライアント・セッションのアイドル・タイムアウトをデフォルト値の 0 秒 (つまり、使用不可) に設定します。
- 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。このコマンドが RMC サブシステムとは別のノード上で実行される場合 (すなわち、**CT_CONTACT** 環境変数が設定されている場合)、このポリシーは、RMC サブシステムと、「*RSCT: 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。タイムアウト値には、以下のアクションを組み込む必要があります。
- 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を受信する。
 - 初期メッセージを受信した後、RMC サブシステムによる完全なクライアント・メッセージを受信する。
- 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をデフォルト値の **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 (**x**) のアクセス権を持つようにする必要があります。

終了状況

0 コマンドは正常に実行されました。

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

/opt/rsct/bin/rmcctrl

rmcdomainstatus コマンド

目的

管理ドメインおよびピア・ドメインにおけるノードの状況を表示します。

構文

```
rmcdomainstatus -s ctrmc [-a ip|IP]
```

説明

ノードにおいて **rmcdomainstatus** コマンドを実行すると、そのノードが含まれる管理ドメインおよびピア・ドメインにおけるノード状況が出力に表示されます。出力が表示されない場合は、対象ノードがピア・ドメインのメンバーでも管理ドメインのメンバーでもありません。

rmcdomainstatus コマンドの出力は次のフォーマットで表示されます。

Domain status

```
<Token 1 of node status> <Token 2 of node status> <Node ID> <Internal node number> <Node name>  
| IP address | IP address of the specified MCP> <PD_name>/<PD_status> (n)
```

rmcdomainstatus コマンド出力には以下の情報フィールドが表示されます。

Domain status

ドメインの現行状況が表示されます。ドメイン状況は以下のように表示される可能性があります。

Peer Domain Status

ピア・ドメインの現行状況が表示されます。

Management Domain Status: Managed Nodes

管理ドメインにおけるノードで管理されているすべての管理対象ノードの現行状態が表示されます。

Management Domain Status: Management Control Points (MCP)

管理ドメインにおけるノードを管理しているすべての管理制御点 (MCP) の現行状態が表示されます。

注: ノードの現行状態によっては、出力に複数のセクションが含まれることがあります。つまり、ノードがピア・ドメインと管理ドメインの両方のメンバーである場合、出力には 2 つの個別の情報セクションが含まれます。

Token 1 of node status

以下のいずれかの状態を示すノード状況が指定されます。

- S** これは、出力内のテキストがピア・ドメインにおける現行ノードに関するものであることを示します。
- I** 管理ドメインの場合、この値はノードが Up 状態にあることを示します。これは Resource Monitoring and Control (RMC) ハートビート・メカニズムによって決定されます。ピア・ドメインの場合、この値は指定のノードにおける RMC デーモンが rmc_peers Group Services グループのメンバーであり、そのノードがピア・ドメインでオンラインになっていることを示します。

- i 管理ドメインの場合、この値はノードが Pending Up 状態にあることを示します。2 つの RMC デーモン間で接続が確立されていますが、初期ハンドシェイクが完了していません。

注: i トークンは管理ドメインに関してのみ表示されます。

- O 管理ドメインの場合、この値はノードが Down 状態にあることを示します。これは RMC ハートビート・メカニズムによって決定されます。ピア・ドメインの場合、この値は指定のノードにおける RMC デーモンが rmc_peers Group Services グループのメンバーではなくなったことを示します。
- X 管理ドメインでは、この値は通信問題がディスカバーされたこと、およびに RMC デーモンが、指定のノードにある RMC デーモンとの通信を中断したことを示します。
- Z これは、指定のノードの Up/Down 状態が素早く変化しているため、RMC デーモンが、そのノードにある RMC デーモンとの通信を中断したことを示します。

Token 2 of node status

以下のいずれかの状態を示すノード状況が指定されます。

- S これは、出力内のテキストがピア・ドメインにおける現行ノードに関するものであることを示します。
- A これは、指定のノードに関するメッセージがキューに入れられていないことを示します。
- a これは、A 値と同じ意味を示します。ただし、指定のノードで実行されている RMC デーモンのバージョンはローカル RMC デーモンよりも低いレベルにあります。
- R これは、指定のノードに関するメッセージがキューに入れられた状態であることを示します。
- r これは、R 値と同じ意味を示します。ただし、指定のノードで実行されている RMC デーモンのバージョンはローカル RMC デーモンよりも低いレベルにあります。

Node ID

64 ビット・ノード ID (RSCT がこのノードにインストールされたときに作成されたもの) が指定されます。

Internal node number

RMC デーモンによって使用されている内部ノード番号が指定されます。

Node name or IP address

RMC サブシステムで識別されるノードの名前が指定されます。

注: この値は、そのノードがピア・ドメインまたは管理ドメインのメンバーである場合にのみ表示されます。

IP address of the specified MCP

指定の MCP について最初に構成された IP アドレスが指定されます。

注: この値は、そのノードが MCP である場合にのみ表示されます。

PD_name/PD_status (n)

-a フラグが使用されていないときに管理対象ノードから受信されたピア・ドメイン名およびピア・ドメイン状況が指定されます。

注: この値は、そのノードが MCP である場合にのみ表示されます。

PD_name 属性はオンライン・メンバーの管理対象ノードを持つピア・ドメインの名前です。

PD_status 属性はピア・ドメインの状況です。管理対象ノードがオフラインの場合、*PD_name*/*PD_status* 属性は *!/-* として設定され、(*n*) 属性は存在しません。ピア・ドメイン状況が管理対象ノードから受信される場合、*PD_name* 属性は *+* として設定されます。*n* 属性は、指定の管理対象ノードをメンバーとして持つピア・ドメインに含まれるオンライン・ノードの数です。

フラグ

-s *ctrmc*

これは RSCT デーモン名を指定します。RMC の場合の RSCT デーモン名は *ctrmc* です。

-a *IP|ip*

これは、対象ノード上に構成されている IP アドレスをリストします。-a フラグで指定できる有効値は以下のとおりです。

IP これは、取得された構成済み IP アドレスをすべてリストします。

ip これは、*ctrmc.srcntbl* ファイル (ピア・ドメインの場合) および *ctrmc.mntbl* ファイル/*ctrmc.mcptbl* ファイル (管理ドメインの場合) に構成されている IP アドレスをリストします。

実装上の固有な条件

このコマンドは AIX、Linux、および他のオペレーティング・システムの Reliable Scalable Cluster Technology (RSCT) ファイルセットに含まれています。

位置

/opt/rsct/bin/rmcdomainstatus

例

1. ピア・ドメインおよび管理ドメインにおけるノード状況を確認するには、次のコマンドを実行します。

```
rmcdomainstatus -s ctrmc
```

出力が表示されない場合は、対象ノードがピア・ドメインのメンバーでも管理ドメインのメンバーでもありません。

対象ノードがピア・ドメインのメンバーである場合は、次の例のような出力が表示されます。

```
Peer Domain Status
 I A 0x09898b3065189db6 0002 test1.ppd.pok.ibm.com
 S S 0x07e7287425d0becd 0001 test2.ppd.pok.ibm.com
```

対象ノードが MCP である場合は、次の例のような出力が表示されます。

```
Management Domain Status: Managed Nodes
 I a 0xbf1fb04e5b7d0b06 0001 test1 !/+
 I a 0x3a75dd6c235c428e 0002 test2 masMMtest/+ (1)
 I A 0x07e7287425d0becd 0003 test3 masfive/+ (2)
 I A 0x09898b3065189db6 0004 test4 masfive/+(2)
```

対象ノードが管理対象ノードである場合は、次の例のような出力が表示されます。

```
Management Domain Status: Management Control Points
 I A 0xef889c809d9617c7 0001 9.xx.xx.xxx
```

2. 取得された構成済み IP アドレスを現行ノード状況で表示するには、次のコマンドを実行します。

```
rmcdomainstatus -s ctrmc -a IP
```

次の例のような出力が表示されます。

```
Peer Domain Status
I A 0x4313b01f7aae13d9 0002 myrsct1.in.ibm.com
S S 0xa15313e0cc675d54 0001 myrsct2.in.ibm.com

Management Domain Status: Management Control Points
I A 0x128a32b77a5d91cb 0001 10.xx.xx.xx (C)
```

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

項目	説明
-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

項目	説明
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. `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

項目	説明
<code>/etc/cifs_fs/cifscred</code>	CIFS 信用証明情報を保管します。

関連情報:

chcifscred コマンド
 mkcifsmnt コマンド
 mkcifscred コマンド
 lscifsmnt コマンド

rmcifsmnt 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

rmcifsmnt -f MountPoint [-B | -N]

説明

rmcifsmnt 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、**rmcifsmnt** コマンドはそれをアンマウントします。

フラグ

項目	説明
-B	対応するエントリーを <code>/etc/filesystems</code> 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。
-f MountPoint	CIFS マウントのパス名を指定します。
-N	ファイルシステムをアンマウントしますが、 <code>/etc/filesystems</code> ファイルからエントリーは除去しません。

終了状況

項目	説明
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。

```
rmcifsmnt -f /mnt
```

位置

`/usr/sbin/rmcifsmnt`

ファイル

項目
`/etc/filesystems`

説明
CIFS エントリーを保管します。

関連情報:

`chcifscred` コマンド
`chcifsmnt` コマンド
`lscifsmnt` コマンド
`mkcifsmnt` 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` ユーザーまたは権限のあるユーザーのみが、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

項目	説明
-d <i>Config_Dir</i>	<i>/etc/wlm/Config_dir</i> 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、 <i>/etc/wlm/current</i> 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 <i>SuperClass</i>	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。 <ol style="list-style-type: none">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。 <code>rmclass options -S Super Sub</code>

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

ファイル

項目	説明
classes	クラスの名前と定義が含まれています。
limits	リソース限度が含まれています。
shares	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

rmcluster コマンド

目的

既存のクラスターまたはサイト構成を除去します。

構文

```
rmcluster [-n clustername] [-S sitename][-v]
```

説明

rmcluster コマンドは、クラスター構成またはクラスター内のいずれかのサイトを除去します。除去する必要があるエンティティーに関連付けられている、リポジトリー・ディスクおよび SAN Volume Controller (SVC) の共用ディスクが解放されます。

あるサイトがクラスターから除去されると、このサイトで使用されていたリポジトリおよび共用ディスクは解放されます。ディスクが解放されてもサイトは除去されません。クラスターが除去されると、すべてのリポジトリおよび共用ディスクが解放されます。

注: サイトが自身を除去することはできません。サイトは、別のサイトにあるノードからしか除去できません。

フラグ

項目	説明
-n <i>clustername</i>	除去するクラスターの名前を指定します。
-S <i>sitename</i>	除去するサイトの名前を指定します。
-v	冗長モードを指定します。

例

1. クラスター構成を除去するには、以下のコマンドを入力します。

```
rmcluster -n mycluster
```

2. *mysite* という名前のサイトをクラスターから除去するには、次のコマンドを別のサイトのノードから入力します。

```
rmcluster -S mysite
```

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・マネージャーを使用することで、このリソース・クラスに対する読み取り権限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・ライン・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・ラインに入力したフラグが誤りです。
- 4 コマンド・ラインに入力したパラメーターが誤りです。
- 5 誤ったコマンド・ライン入力により、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

Resource Monitoring and Control (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 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 **ApplDomain** に定義され、オンラインになっています。ピア・ドメイン **ApplDomain** 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

```
/opt/rsct/bin/rmcomg
```

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 環境変数が決定する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・ライン・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・ラインに入力したフラグが誤りです。
- 4 コマンド・ラインに入力したパラメーターが誤りです。
- 5 誤ったコマンド・ライン入力により、エラーが起きました。

環境変数

CT_CONTACT

Resource Monitoring and Control (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

/opt/rsct/bin/rmcondition

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。このため、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

condition

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。

response

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。

node_name

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 環境変数が決定する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・ライン・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・ラインに入力したフラグが誤りです。
- 4 コマンド・ラインに入力したパラメーターが誤りです。
- 5 誤ったコマンド・ライン入力により、エラーが起きました。

環境変数

CT_CONTACT

Resource Monitoring and Control (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

/opt/rsct/bin/rmcondresp

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

項目	説明
-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	rmcosi 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

項目	説明
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

項目
/etc/niminfo

説明
NIM が使用する変数が入っています。

関連資料:

104 ページの『nim_clients_setup コマンド』

109 ページの『nim_master_setup コマンド』

144 ページの『nimconfig コマンド』

関連情報:

chcosi コマンド

cpcosi コマンド

rmdel コマンド

目的

SCCS ファイルからデルタを除去します。

構文

```
rmdel -r SID File ...
```

説明

rmdel 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、**rmdel** 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、**rmdel** 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。**rmdel** 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

項目	説明
<code>-r SID</code>	SCCS ファイルから指定のデルタ <i>SID</i> 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。

項目	説明
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

項目	説明
<code>/usr/bin/rmdev</code>	<code>rmdev</code> コマンドが入っています。
<code>s.files</code>	<code>rmdev</code> 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

ソース・コード制御システム (SCCS) の概要

SCCS コマンドのリスト

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 ] [ -g ]
```

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

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。

このコマンドは、System Management Interface Tool (SMIT)smit rmdev 高速パスを使用して実行することができます。

フラグ

項目	説明
-d	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 -S フラグとともに使用できません。
-f <i>File</i>	<i>File</i> パラメーターから必要なフラグを読み取ります。
-g	ロックしたデバイスでの削除操作の実行を強制します。
-h	コマンド使用方法メッセージを表示します。
-I <i>Name</i>	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 -p フラグとともに使用することはできません。
-p <i>Name</i>	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 -I フラグとともに使用できません。
-q	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。
-R	デバイスおよびその子デバイスを構成解除するよう指定します。-d または -S 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。
-S	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してデバイスを使用不可にします。このフラグは -d 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。

項目	説明
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
DEV_Stop	デバイス名。
DEV_Unconfigure	デバイス名。
DEV_Remove	デバイス名。

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
<code>/usr/sbin/rmdev</code>	<code>rmdev</code> 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

`lsparent` コマンド

`mkdev` コマンド

rmmdir コマンド

目的

ディレクトリーを除去します。

構文

`rmmdir [-p] Directory ...`

説明

`rmmdir` 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: `rmmdir` 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

項目	説明
<code>-pDirectory</code>	<code>Directory</code> 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。

項目	説明
0	<code>Directory</code> 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/.  
rmmdir 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

項目	説明
/usr/bin/rmdir	rmdir コマンドが入っています。

関連情報:

ファイル・コマンド

ディレクトリー・コマンド

ls コマンド

mkdir コマンド

unlink コマンド

rmdom コマンド

目的

ドメイン・データベースからドメインを除去します。

構文

rmdom *Name*

説明

rmdom コマンドは、*Name* パラメーターによって識別されるドメインを除去します。このコマンドは、ドメイン・データベースから既存のドメインのみを除去します。ドメイン・オブジェクト・データベースによって参照されるドメインは、そのドメインへの参照を除去するまでは除去できません。

システムが拡張された役割ベースのアクセス制御 (RBAC) モードで作動している場合、ドメイン・データベースに対して行われた変更は、**setkst** コマンドを使用してデータベースがカーネル・セキュリティー・テーブルに送られるまで、セキュリティー上の考慮事項としては使用されません。

パラメーター

項目 名前	説明 除去するドメインの名前を指定します。
----------	--------------------------

セキュリティ

rmdom コマンドは特権コマンドです。このコマンドを実行するためには、以下の権限が必要です。

項目 aix.security.domains.remove	説明 ドメイン・データベースからドメインを除去するために必要です。
--	--------------------------------------

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

アクセスされるファイル

ファイル /etc/security/domains	モード rw
--------------------------------------	-----------

例

hrdom ドメインを除去するには、次のように入力します。

```
rmdom hrdom
```

関連情報:

mksdom コマンド

chdom コマンド

lsdom コマンド

/etc/security/domain コマンド

RBAC コマンド

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

項目	説明
+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	フォルダー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ー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。

項目	説明
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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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. 現行フォルダー **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"?
```

フォルダー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** フォルダー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

項目	説明
<code>\$HOME/mh_profile</code>	MH ユーザー・プロファイルを定義します。
<code>/usr/bin/rmf</code>	<code>rmf</code> コマンドが入っています。

関連資料:

817 ページの『`rmm` コマンド』

関連情報:

`mh_alias` コマンド

`mh_profile` コマンド

メール・アプリケーション

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・ルールは、`genfilt` コマンド、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 IPsec smit (IP バージョン 4 または IP バージョン 6) 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

項目	説明
<code>-f</code>	自動生成フィルター規則を強制的に除去します。 <code>-f</code> フラグを <code>-n all</code> 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。
<code>-n</code>	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。 <code>all</code> を指定すると、 <code>-f</code> 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。
<code>-v</code>	除去する必要が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。

セキュリティ

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権について詳しくは、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストに

については、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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。

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

項目 説明

- 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。

項目	説明
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。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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ー、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

項目	説明
/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。

関連資料:

814 ページの『**rmlv** コマンド』

関連情報:

chfs コマンド

mkfs コマンド

ファイルシステム

System Management Interface Tool

rmgroup コマンド

目的

グループを除去します。

構文

```
rmgroup [-p] [ -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。

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

項目	説明
-p	グループ鍵ストアを除去します。
-R <i>load_module</i>	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。

項目	説明
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。障害のタイプの詳細については、出力されたエラー・メッセージを参照してください。

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
r	/etc/passwd
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
GROUP_Remove	グループ

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

制限

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

項目	説明
<code>/usr/sbin/rmggroup</code>	rmgroup コマンドが入っています。
<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。

関連資料:

312 ページの『`passwd` コマンド』

865 ページの『`rmuser` コマンド』

関連情報:

`chfn` コマンド

`mkuser` コマンド

ネットワークの保護

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

項目	説明
-g <i>group</i>	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 <i>IPAddress</i>	iSCSI ターゲットの IP アドレスを指定します。
-l <i>AdapterName</i>	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 <i>PortNumber</i>	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。

項目	説明
<code>-t TargetName</code>	iSCSI ターゲット名 (例えば、 <code>iqn.sn9216.iscsi-hw1</code>) を指定します。

終了状況

項目	説明
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 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
```
- 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

項目	説明
<code>src/bos/usr/sbin/iscsia</code>	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

`lsiscsi` コマンド

`mkiscsi` コマンド

rmitab コマンド

目的

`/etc/inittab` レコード内の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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、`lssecattr` コマンドまたは `getcmdattr` サブコマンドの項を参照してください。

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

init コマンド

/etc/inittab コマンド

rmkeyserd コマンド

目的

keyserd 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/rmkeyserd [ -I | -B | -N ]
```

説明

rmkeyserd コマンドは、**/etc/rc.nfs** ファイルの **keyserd** デーモンに関するエントリーにコメントを付けます。**rmkeyserd** デーモンは、**stopsrc** コマンドを使用して **keyserd** デーモンを停止します。

また、System Management Interface Tool (SMIT) **smit rmkeyserd** 高速パスを使用して実行することもできます。

フラグ

項目	説明
-I	/etc/rc.nfs ファイルの keyserd 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 keyserd デーモンに関するエントリーにコメントを付け、 keyserd 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 keyserd 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

keyserd 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。

```
rmkeyserd -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

項目	説明
<code>/etc/rc.nfs</code>	NFS デーモンと NIS デーモンの起動スクリプトが入っています。

関連情報:

smit コマンド

keyserv コマンド

システム管理のためのネットワーク・ファイルシステム (NFS) の概要、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

Network Information Service (NIS)

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。

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

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。

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

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 **lpacl** 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・ライン・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・ラインに間違っただフラグが指定されました。
- 4 コマンド・ライン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・ライン入力に基づく 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 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。

```
rmlpcmd 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。

```
rmlpcmd LP1 LP2
```

位置

/opt/rsct/bin/rmlpcmd

rmlpcmd コマンドが入っています。

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 ユーザー権限を持っているか、あるいは **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。
3. **rmlv** コマンドは、アクティブなファームウェア支援ダンプ論理ボリュームには使用できません。
4. AIX 7.2 テクノロジー・レベル 1 以降では、区画は、解放された区画に対する **rmlv** コマンドを実行すること、すなわちスペース再利用プロセスの実行によって解放されます。

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

項目	説明
-B	論理ボリュームが -l 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	<i>PhysicalVolume</i> 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。

セキュリティ

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
<code>/usr/sbin/rmlv</code>	rmlv コマンドが常駐するディレクトリー。
<code>/tmp</code>	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。
<code>/etc/filesystems</code>	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。

関連資料:

806 ページの『**rmfs** コマンド』

関連情報:

varyonvg コマンド

unmount コマンド

論理ボリューム・ストレージ

System Management Interface Tool

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。

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・ライン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・ラインで指定され、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。
5. **rmlvcopy** コマンドをアクティブなファームウェア・アシストのダンプ論理ボリューム上で実行すると、ダンプ・デバイスを **/dev/sysdumpnull** ファイルに一時的に変更します。この論理ボリューム・コピーが正常に削除された後、**rmlvcopy** コマンドは **sysdumpdev -P** コマンドを呼び出して、ファームウェア支援ダンプ論理ボリュームを元のダンプ論理ボリュームに設定します。
6. AIX 7.2 テクノロジー・レベル 1 以降では、区画は、解放された区画に対する **rmlvcopy** コマンドを実行すること、すなわちスペース再利用プロセスの実行によって解放されます。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
/usr/sbin/rmlvcopy	rmlvcopy コマンドが入っています。
/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:

mklv コマンド

mklvcopy コマンド

論理ボリューム・ストレージ

System Management Interface Tool

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

項目	説明
+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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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。

項目	説明
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。
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

項目	説明
\$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。

関連資料:

803 ページの『**rmf** コマンド』

関連情報:

crontab コマンド

.mh_alias コマンド

メール・アプリケーション

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ーを構成解除するか、**/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。

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

項目	説明
-F <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。
-f	/etc/resolv.conf 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 (/etc/resolv.conf.sv) を使うことを指定します。

ファイル

項目	説明
/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。

関連資料:

9 ページの『**namerslv** コマンド』

関連情報:

naming 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

項目	説明
-B	inittab ファイル内のエントリーを除去して、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ーを除去します。
-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。

関連資料:

822 ページの『**rmnfsmnt** コマンド』

関連情報:

chnfs コマンド

mknfs コマンド

NFS コマンドのリスト

ネットワーク・ファイルシステム (NFS) の概要

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ーを除去します。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ーが存在する場合は、そのエントリーが除去されます。

フラグ

項目	説明
-d <i>Directory</i>	アンエクスポートしたいディレクトリーを指定します。
-f <i>Exports_File</i>	/etc/exports 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。
-I	コマンドに対して、 exportfs コマンドを実行せずに /etc/exports 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	指定されたディレクトリーのエントリーを /etc/exports ファイルで除去し、 exportfs コマンドを実行してエクスポートを除去します。
-N	exportfs 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、 /etc/exports ファイルは修正されません。
-V <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。
-F	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

項目	説明
<code>/etc/xtab</code> <code>html</code>	現在エクスポートされているディレクトリーのリスト。

関連情報:

`chnfsexp` コマンド

`exportfs` コマンド

`mknfsexp` コマンド

NFS コマンドのリスト

ネットワーク・ファイルシステム (NFS) の概要

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 [ -I | -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

項目	説明
-B	<code>/etc/filesystems</code> 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。 <code>/etc/filesystems</code> 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ければ、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。
-I	パス名で指定されたエントリーを <code>/etc/filesystems</code> ファイルから除去します。
-N	<code>/etc/filesystems</code> 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

項目	説明
<code>/etc/filesystems</code>	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

`umount` コマンド

ネットワーク・ファイルシステム (NFS) の概要

NFS コマンドのリスト

rmnfsproxy コマンド

目的

プロキシが使用可能化されている `Cachefs` 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

項目	説明
<code>Cachefs_mount_point</code>	除去対象の、プロキシが使用可能化された <code>Cachefs</code> 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

項目	説明
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

- 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

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

項目	説明
-n <i>NotifyName</i>	除去する通知メソッド定義を指定します。 <i>NotifyName</i> 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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**Issecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

ファイル

項目	説明
/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

システム・リソース・コントローラー

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

SRC オブジェクトについて

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*] [-i *PathID*]

rmpath [-l *Name*] [-p *Parent*] [-w *Connection*] [-d] [-g]

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

項目

-d

-g

-h

-i PathID

-l Name

説明

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。

ロックしたデバイスでのパス削除操作の実行を強制します。

コマンド使用方法メッセージを表示します。

除去するパスに関連付けられたパス ID を示し、パスを一意的に識別するために使用されます。

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、**-p** および **-w** フラグを使用して修飾されます。

項目	説明
-p Parent	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 -w 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。
-w Connection	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 -p 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

項目
/usr/sbin/rmpath

説明
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** サーバーを非活動状態にします。
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

項目	説明
-A	指定されたエントリーを /etc/qconfig 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。
-c	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。 -q フラグを -c フラグとともに使ってください。
-H FileName	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。
-h "HostName..."	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。
-q "QEntry..."	/etc/qconfig 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。
-s	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。 -s フラグとともに、 -h 、 -H 、および -q フラグを使わなければなりません。
-T	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。
-U	指定されたリモート・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。

項目	説明
<code>-v "DeviceName..."</code>	qconfig 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。 <code>-q QEntry</code> フラグとともに使用してください。

ファイル

項目	説明
<code>/etc/qconfig</code>	プリンター・キュー・システムの構成情報が入っています。

関連資料:

831 ページの『`rmquedev` コマンド』

931 ページの『`ruser` コマンド』

592 ページの『`qdaemon` コマンド』

関連情報:

`qconfig` ファイル

`lpd` コマンド

rmmps 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

rmmps[`-t ps_helper`] *PagingSpace*

説明

rmmps 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、除去する必要があるページング・スペースの名前を指定します。このページング・スペースは、ページング・スペースが存在する論理ボリュームの名前です。

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。ページングに使用されるファイルが存在する 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、実行可能モジュールは、**rmmps** コマンドを指定するために `-r` フラグを引き渡すことによって作成されます。ヘルパー実行可能モジュールがゼロを返すように、`/etc/swapspace` ディレクトリーが変更されます。

ヘルパー実行可能モジュールは、ページング・スペースの除去に使用されます。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、**rmmps** コマンドは使用法のエラーを表示します。ヘルパー実行可能モジュールは、正常実行された場合は値 `0` で、失敗した場合はゼロ以外の値で終了します。

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。

フラグ

項目 説明

-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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権命令を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権についての詳細情報は、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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。

```
rmips 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** 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。

```
rmips -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

項目

/etc/swapspace

説明

ページング・スペース・デバイスとその属性を指定します。

関連情報:

swapoff コマンド

chps コマンド

ファイルシステム

論理ボリューム・ストレージ

qconfig ファイル

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

項目	説明
-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。

セキュリティ

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

ファイル

項目	説明
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

TCP/IP サービス品質 (QoS)

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・スタンザを削除して、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。

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

項目	説明
<code>-q Name</code>	除去するキューの名前を指定します。

セキュリティ

RBAC ユーザーおよび **Trusted AIX ユーザー**への注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、`lssecattr` コマンドまたは `getcmdattr` サブコマンドの項を参照してください。

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

項目	説明
<code>/usr/bin/rmque</code>	<code>rmque</code> コマンドが入っています。
<code>/etc/qconfig</code>	構成ファイルが入っています。

関連情報:

`qconfig` ファイル

`lsque` コマンド

印刷の管理

印刷キューの削除

rmquedev 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

```
rmquedev -d Name -q Name
```

説明

`rmquedev` 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。

また、System Management Interface Tool (SMIT) `smit rmquedev` 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`、`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

項目	説明
<code>-d Name</code>	<code>qconfig</code> ファイルから削除するデバイス・スタンザの <code>Name</code> を指定します。
<code>-q Name</code>	前のキュー・スタンザで修正するデバイスの <code>Name</code> を指定します。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、`lssecattr` コマンドまたは `getcmdattr` サブコマンドの項を参照してください。

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。

```
rmquedev -q lpq -d loc
```

ファイル

項目	説明
<code>/usr/bin/rmquedev</code>	<code>rmquedev</code> コマンドが入っています。
<code>/etc/qconfig</code>	構成ファイル。

関連情報:

`chquedev` コマンド

`lsquedev` コマンド

`/etc/qconfig` コマンド

`qconfig` ファイル

印刷キューの削除

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

項目	説明
<code>ram_disk_name</code>	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、 <i>x</i> 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。

項目	説明
0	正常終了。
>0	エラーが発生しました。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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。

```
# rmramdisk ramdisk2

# ls -l /dev/*ramdisk2
ls: 0653-341 The file /dev/*ramdisk2 does not exist.
```

ファイル

項目	説明
<code>/usr/sbin/rmramdisk</code>	rmramdisk コマンドが入っています。

関連情報:

`mkramdisk` 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 環境変数によって決定される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・ライン・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・ラインに入力したフラグが誤りです。
- 4 コマンド・ラインに入力したパラメーターが誤りです。
- 5 誤ったコマンド・ライン入力により、エラーが起きました。

環境変数

CT_CONTACT

Resource Monitoring and Control (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

/opt/rsct/bin/rmresponse

rmrole コマンド

目的

ロールを除去します。

構文

```
rmrole [-R load_module] 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。

rmrole コマンドは、System Management Interface Tool (SMIT) を使用して実行することができます。

複数のドメインからデータベースを使用するようにシステムが構成されている場合、**rmrole** コマンドは、**/etc/nscontrol.conf** ファイル内の対応するロール・スタンザの **secorder** 属性で指定された順序で、データベース・ドメインから最初の一致を検索します。それと同時に、**rmrole** コマンドは、検索されたロールのエントリーをドメインから除去します。残りのドメインに一致するロールがあっても、それらのロールは影響されません。特定のドメインからロールを除去するには、**-R** フラグを使用します。

システムが拡張された役割ベースのアクセス制御 (RBAC) モードで作動している場合、ロール・データベースから除去されたロールは、**setkst** コマンドを使用してカーネル・セキュリティ・テーブル (KST) が更新されるまで、KST 内に存在します。

フラグ

項目	説明
-R <i>load_module</i>	ロールの削除に使用するロード可能モジュールを指定します。

セキュリティ

rmrole コマンドは特権コマンドです。このコマンドを実行するためには、**aix.security.role.remove** 権限が必要です。

項目	説明
<code>aix.security.role.remove</code>	このコマンドの実行に必要です。

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
<code>rw</code>	<code>/etc/security/roles</code>
<code>r</code>	<code>/etc/security/user.roles</code>

監査イベントは次のとおりです。

イベント	情報
<code>ROLE_Remove</code>	ロール

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティー」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、`lssecattr` コマンドまたは `getcmdattr` サブコマンドの項を参照してください。

例

1. `ManageObjects` ロールを除去するには、次のコマンドを入力します。

```
rmrole ManageObjects
```
2. `LDAP` から `ManageRoles` ロールを除去するには、次のコマンドを入力します。

```
rmrole -R LDAP ManageRoles
```

ファイル

項目	説明
<code>/etc/security/roles</code>	ロールの属性が入っています。
<code>/etc/security/user.roles</code>	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

ネットワークの保護

ユーザー、ロール、およびパスワード

`RBAC` コマンド

`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。

クラスター対応 AIX (CAA) クラスターが構成済みでピア・ドメインがこれを表している場合、**rmrpdomain** コマンドを使用すると、基本となる CAA クラスターも除去されます。

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・ドメインのどのノード上の **root** も、構成リソース・マネージャーを使用することで、このリソース・クラスに対する読み取り権限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・ライン・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・ラインに入力したフラグが誤りです。
- 4 コマンド・ラインに入力したパラメーターが誤りです。
- 5 誤ったコマンド・ライン入力により、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

Resource Monitoring and Control (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 **rsct.basic.rte** 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** (ここには **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** (ここには **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

/opt/rsct/bin/rmrpdomain

ファイル

/etc/services ファイルが変更されます。

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

```
rmrpnode [-f] [-q] [-h] [-TV] node_name1 [node_name2 ...]
```

```
rmrpnode -F { file_name | "-" } [-f] [-q] [-h] [-TV]
```

説明

rmrpnode 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、**stoprpnode** コマンドを使用します。

クラスター対応 AIX (CAA) クラスターが構成済みでピア・ドメインがこれを表している場合、**rmrpnod**e コマンドを使用すると、基本となる CAA クラスターからノードも除去されます。

-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。**rmrpnod**e **-f** を使用して最後のタイブレーカー・ノードが除去されると、(全ノードではなく) 残りのクォラム・ノードだけがタイブレーカー・ノードになるように変換されます。

このコマンドを実行するときに **-f** フラグを指定しない場合、以下のようにします。

- ドメインから 1 つ以上のノードを除去するには、クォラム・ノードの過半数がオンラインである必要があります。
- 結果的に、残りのタイブレーカー・ノードがピア・ドメインにない場合、エラーが戻されます。

クォラム・ノードとタイブレーカー・ノードの詳細については、「*Administering RSCT*」を参照してください。

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。

このフラグを使用して最後のタイブレーカー・ノードが除去されると、(全ノードではなく) 残りのクォラム・ノードだけがタイブレーカー・ノードになるように変換されます。

クォラム・ノードとタイブレーカー・ノードの詳細については、「*Administering RSCT*」を参照してください。

-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、入力ファイルとして **STDIN** 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* ...]

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。**rmrpnod**e 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。ノード名は、**addrpnod**e コマンドまたは **mkrpdomain** コマンドで指定したものとまったく同じフォーマットで指定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、**lsrpnod**e コマンドを実行します。

セキュリティ

rmrpnode 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・ドメインのどのノード上の **root** も、構成リソース・マネージャーを使用することで、このリソース・クラスに対する読み取り権限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・ライン・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・ラインに入力したフラグが誤りです。
- 4 コマンド・ラインに入力したパラメーターが誤りです。
- 5 誤ったコマンド・ライン入力により、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

Resource Monitoring and Control (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 **rsct.basic.rte** ファイルセットの一部です。

標準入力

このコマンドは、**-F "-"** フラグが指定されている場合は、標準入力から 1 つ以上のノード名を読み取ります。

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 **ApplDomain** から削除します。この時、**nodeA** は **ApplDomain** 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

```
/opt/rsct/bin/rmrpnode
```

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

項目	説明
rsetname	システム・レジストリーから除去する rset の名前。名前は、 <i>namespace</i> および "/ (スラッシュ) で区切られた <i>rsname</i> からなります。 <i>namespace</i> および <i>rsname</i> 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。

セキュリティー

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。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権について詳しくは、「セキュリティー」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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

項目	説明
<code>/usr/bin/rmrset</code>	<code>rmrset</code> コマンドが入っています。

関連情報:

`attachrset` コマンド
`detachrset` コマンド
`execrset` コマンド
`lsrset` コマンド
`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、以下のように実行します。

- コマンド・ライン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 (Resource Monitoring and Control) 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。

-l 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。
- ノード名の左の空白文字はすべて無視されます。
- ノード名の右の空白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。

-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 : **"0xnxxxx 0xnxxxx 0xnxxxxxxxx 0xnxxxxxxxx 0xnxxxxxxxx 0xnxxxxxxxx"** (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・ライン・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・ラインに間違っただフラグが指定されました。
- 4 コマンド・ライン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・ライン入力に基づく 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 (Resource Monitoring and Control) 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 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.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
```

2. 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"
```

3. **/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

/opt/rsct/bin/rmrsrc

rmsecattr コマンド

目的

コマンド、デバイス、特権ファイル、またはドメイン割り当て済みオブジェクトについて、データベース内にあるセキュリティ属性定義を除去します。

構文

```
rmsecattr [-R load_module] { -c | -d | -f | -o } Name
```

説明

rmsecattr コマンドは、コマンド、デバイス、ファイル・エンタリー、または *Name* パラメーターで識別されるドメイン割り当て済みオブジェクトのセキュリティ属性を、該当するデータベースから除去します。このコマンドは、フラグとして **-c** (コマンド)、**-d** (デバイス)、**-f** (特権ファイル)、または **-o** (ドメイン割り当て済みオブジェクト) のいずれかが指定されているかに基づいて、*Name* パラメーターをコマンド、デバイス、ファイル・エンタリー、またはドメイン割り当て済みオブジェクトのいずれかとして解釈します。**-c** フラグを指定する場合、*Name* パラメーターにはコマンドへの絶対パスを含める必要があります、またそのコマンドはその時点で **/etc/security/privcmds** 特権コマンド・データベース内にエンタリーを持っている必要があります。

-d フラグを指定する場合、*Name* パラメーターにはデバイスへの絶対パスを含める必要があります、またそのデバイスはその時点で **/etc/security/privdevs** 特権デバイス・データベース内にエンタリーを持っている必要があります。

-f フラグを指定する場合、*Name* パラメーターにはファイルへの絶対パスを含める必要があります、またそのファイルは **/etc/security/privfiles** 特権ファイル・データベース内にエンタリーを持っている必要があります。

-o フラグを指定する場合、オブジェクト・タイプがファイルまたはデバイスであれば *Name* パラメーターには絶対パスを含める必要があります、またそのオブジェクト・タイプは **/etc/security/domobjs** ドメイン割り当て済みオブジェクト・データベース内にエンタリーを持っている必要があります。

重要: **rmsecattr** コマンドはセキュリティ属性の定義のみを除去します。実際のコマンド、デバイス、またはファイルを除去するものではありません。

複数のドメインからデータベースを使用するようにシステムが構成されている場合、**rmsecattr** コマンドは、**/etc/nscontrol.conf** ファイル内の対応するデータベース・スタanzasの **secorder** 属性で指定された順序で、データベース・ドメインから最初の一致を検索します。それと同時に、**rmsecattr** コマンドは、検索されたコマンドまたはデバイスのエンタリーをドメインから除去します。残りのドメインに一致するエンタリーがあっても、それらのエンタリーは影響されません。特定のドメインからエンタリーを除去するには、**-R** フラグを使用します。

このコマンドで行われた変更は、**setkst** コマンドを使用してデータベースがカーネル・セキュリティ・テーブルに送られるまで、セキュリティ上の考慮事項としては使用されません。

フラグ

項目	説明
-c	<i>Name</i> パラメーターと一緒に使用した場合、特権コマンド・データベースにエントリーを持つシステム上の 1 つ以上のコマンドへの絶対パスを指定します。
-d	<i>Name</i> パラメーターと一緒に使用した場合、特権デバイス・データベースにエントリーを持つシステム上の 1 つ以上のデバイスへの絶対パスを指定します。
-f	<i>Name</i> パラメーターと一緒に使用した場合、システム上の特権ファイルへの絶対パスを指定します。
-o	<i>Name</i> パラメーターと一緒に使用した場合、ドメイン割り当て済みオブジェクト・データベースで指定されたオブジェクトを指定します。
-R <i>load_module</i>	<i>Name</i> エントリーの削除に使用するロード可能モジュールを指定します。

パラメーター

項目 名前	説明
	変更するオブジェクト。 <i>Name</i> パラメーターは、指定された -c 、 -d 、 -f 、または -o フラグに従って解釈されます。

セキュリティ

rmsecattr コマンドは特権コマンドです。このコマンドはモードが 755 に設定され、root ユーザーとセキュリティ・グループによって所有されています。このコマンドを実行するには、次の権限の 1 つ以上が必要です。

項目	説明
aix.security.cmd.remove	-c フラグを指定してコマンドのセキュリティ属性を除去するために必要です。
aix.security.device.remove	-d フラグを指定してデバイスのセキュリティ属性を除去するために必要です。
aix.security.object.remove	-o フラグを指定してドメイン割り当て済みオブジェクトのセキュリティ属性を除去するために必要です。
aix.security.file.remove	-f フラグを指定してファイルのセキュリティ属性を除去するために必要です。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

アクセスされるファイル

ファイル	モード
/etc/security/domobjs	rw
/etc/security/privcmds	rw
/etc/security/privdevs	rw
/etc/security/privfiles	rw

例

1. 特権コマンド・データベースから **/usr/sbin/mytest** コマンドを除去するには、次のように入力します。

```
rmsecattr -c /usr/sbin/mytest
```
2. 特権デバイス・データベースから **/dev/mydev** デバイスを除去するには、次のように入力します。

```
rmsecattr -d /dev/mydev
```

- LDAP の特権デバイス・データベースから `/dev/mydev` デバイスを除去するには、次のように入力します。

```
rmsecattr -R LDAP -d /dev/mydev
```

- 特権ファイル・データベースから `/etc/testconf` ファイルを除去するには、次のように入力します。

```
rmsecattr -f /etc/testconf
```

- ドメイン割り当て済みオブジェクト・データベースからネットワーク・インターフェース `en0` を除去するには、次のように入力します。

```
rmsecattr -o objecttype=netint en0
```

関連情報:

pvi コマンド

getcmdattr コマンド

getcmdattrr コマンド

`/usr/lib/security/methods.cfg` コマンド

RBAC コマンド

rmSENSOR コマンド

目的

センサーまたはマイクロセンサーを Resource Monitoring and Control (RMC) サブシステムから除去します。

構文

```
rmSENSOR [ -m ] [-a | -n host1[,host2...] | -N { node_file | "-" } ] [-h] [-v | -V] sensor_name1  
[sensor_name2...]
```

説明

rmSENSOR コマンドは RMC サブシステムの **IBM.Sensor** リソース・クラスから 1 つ以上のセンサーを除去したり、**IBM.MicroSensor** リソース・クラスから 1 つ以上のマイクロセンサーを除去したりします。マイクロセンサーの除去には **-m** フラグを使用します。

センサーまたはマイクロ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ている、イベント応答のリソース・マネージャー (ERRM) リソースは除去されません。ERRM リソースを除去するには、このセンサーまたはマイクロセンサーで使用していたモニター中のリソースに対して、コマンド **rmcondition**、**rmresponse**、または **rmcondresp** 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。**-n** フラグを使用して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内にあることを指示することができます。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** 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。
- m** 除去するリソースがマイクロセンサー・リソースであることを指定します。
- 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。
- 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

このコマンドを使用してセンサーを除去するには、**IBM.Sensor** リソース・クラスに対する書き込み権限が必要です。このコマンドを使用してマイクロセンサーを除去するには、**IBM.MicroSensor** リソース・クラスに対する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 (Resource Monitoring and Control) 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 rsct.core 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
2. **/u/joe/common_nodes** ファイルにリストされているノードから **sensor1** というセンサーをノードから除去するには、次のように入力します。

```
rmsensor -N /u/joe/common_nodes sensor1
```

この **/u/joe/common_nodes** には、以下のものが含まれます。

```
# common node file
#
node1.myhost.com    main node
node2.myhost.com    backup node
```

3. **IBM.usensor1** というマイクロセンサーを除去するには、次のように入力します。

```
rmsensor -m IBM.usensor1
```

位置

/opt/rsct/bin/rmsensor

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

項目	説明
-t <i>Type</i>	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。 <i>Type</i> 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
SRC_Delserver	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。

html を参照してください。

関連情報:

システム・リソース・コントローラー

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

auditpr コマンド

startsrc コマンド

stopsrc コマンド

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。一致が見つかったら、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/rmssock.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、 **rmssock** 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。

```
rmssock 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。

2. 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。

```
rmssock 70054edc inpcb
```

3. 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。

```
rmssock 70054ecc tcpcb
```

ファイル

項目	説明
/usr/sbin	rmssock コマンドが常駐しているディレクトリー。
/var/adm/ras/rmssock.log	rmssock.log ファイルが含まれています。

関連情報:

syslogd コマンド

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 ] コマンド
```


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・ライン引数は指定してもしなくてもかまいません。

rmss コマンドは、DR サブシステムと非互換です。**rmss** コマンドの実行中に DR イベントが発生すると、システムがハングすることがあります。**rmss** コマンドのメモリー除去機能は、**drmgr** コマンドでの DR メモリー除去によって置き換えることができるため、**rmss** コマンドの情報テキストを修正して、以下の注意を含める必要があります。

重要: **rmss** コマンドは AIX DLPAR コンポーネントと非互換であり、これを使用するとシステムがハングする可能性があります。**drmgr** コマンドは、DLPAR 環境に安全なメモリー除去機能を提供します。

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す。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」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. デー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
32	1000
48	1500
64	2000
128	4000
256	8000

あるコマンドが多く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

項目	説明
-c MemSize	シミュレートされるメモリーのサイズを <i>MemSize</i> 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。 <i>MemSize</i> 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。 <i>MemSize</i> 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 (<i>MemSize</i> 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。 <i>MemSize</i> 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。

項目	説明
-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 (<i>MemSize</i> 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。 <i>MemSize</i> 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。
コマンド	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。 <i>Command</i> 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・ライン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。

項目	説明
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。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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
```

- 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
```

- 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
```

- 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

項目	説明
<code>/usr/bin/rmss</code>	rmss コマンドが入っています。

関連情報:

filemon コマンド

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

項目	説明
-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「 」の場合、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』を参照してください。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

ファイル

項目	説明
/etc/objrepos/SRCsubsys	SRC サブシステム構成オブジェクト・クラスを指定します。
/etc/objrepos/SRCsubsvr	SRC Subserver Configuration オブジェクト・クラスを指定します。
/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。
/dev/SRC	AF_UNIX ソケット・ファイルを指定します。
/dev.SRC-unix	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

システム・リソース・コントローラー

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

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

項目	説明
ODeviceMode	Mode パラメーターで指定したモードを使用して、 Device パラメーターで指定したデバイスをオープンします。 Device パラメーターの値はフル・パス名です。また、 Mode パラメーターの値は、 open 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。
CDevice	オープン・デバイスをクローズします。 Device パラメーターで指定したデバイスは無視されます。
LWhenceOffset	指定されたパラメーターを使用して、 lseek 操作を行います。 lseek サブルーチンは応答値を戻します。
WCount	データをオープン・デバイスに書き込みます。 rmt コマンドは接続から、 Count 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があると終了します。 write サブルーチンは応答値を戻します。

項目	説明
RCount	Count 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、 rmt 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。
IOperationCount	指定したパラメーターを使用して、 STIOCTOP ioctl 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、ioctl サブルーチンで使われる構造の mt op および mt count フィールドに配置されます。操作が終了すると、 Count 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。

項目	説明
0	正常終了。
>0	エラーが発生しました。

ファイル

項目	説明
/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。

関連資料:

- 670 ページの『**rdump** コマンド』
- 899 ページの『**rrestore** コマンド』
- 750 ページの『**rexec** コマンド』

関連情報:

rmt コマンド

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。

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

項目	説明
<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

項目	説明
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

項目	説明
/etc/niminfo	NIM が使用する変数が入っています。

関連資料:

88 ページの『**nim** コマンド』

104 ページの『**nim_clients_setup** コマンド』

関連情報:

dbts コマンド

lsts コマンド

mkts 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

項目	説明
all	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。
tid_list	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。
-d	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。
-t	活動停止したいトンネルのリスト。 -d 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。
-v	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、 4 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、 6 の値を使用してください。

セキュリティ

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

関連情報:

chtun コマンド

exptun コマンド

gentun コマンド

lstun コマンド

mktun コマンド

rmusil コマンド

目的

既存のユーザー指定インストール・ロケーション (USIL) インスタンスを除去します。

構文

rmusil -R RelocatePath -r

説明

rmusil コマンドは既存の USIL インスタンスを除去します。

フラグ

項目	説明
-r	USIL インスタンスのソフトウェア重要プロダクト・データ (SWVPD) を除去します。
-R RelocatePath	既存の USIL ロケーションへのパス。

注: **rmusil** コマンドは、ソフトウェア SWVPD の中の USIL 参照のみを除去します。USIL インストール・パスのファイルは除去されません。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

ファイル

項目	説明
<code>/usr/sbin/rmusil</code>	rmusil コマンドが入っています。

関連情報:

chusil コマンド

lsusil コマンド

mkusil 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。このコマンドはユーザー・アカウントの属性を除去しますが、ユーザーのホーム・ディレクトリーとファイルは除去しません。ユーザー名は、存在していなければなりません。 **-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その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・アカウントの場合は、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。

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。

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

項目	説明
-P	<code>/etc/security/passwd</code> ファイルからユーザー・パスワード情報を除去し、さらにユーザー鍵ストアを除去します。
-R <code>load_module</code>	ユーザー・アカウント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。

パラメーター

項目 名前	説明
	ユーザー・アカウントを指定します。

終了状況

このコマンドは、以下の終了値を戻します。

項目	説明
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。障害のタイプの詳細については、出力されたエラー・メッセージを参照してください。

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	<code>/etc/passwd</code>
rw	<code>/etc/security/passwd</code>
rw	<code>/etc/security/user</code>
rw	<code>/etc/security/user.roles</code>
rw	<code>/etc/security/limits</code>
rw	<code>/etc/security/environ</code>
rw	<code>/etc/security/audit/config</code>
rw	<code>/etc/group</code>
rw	<code>/etc/security/group</code>

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

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

1. ユーザー・アカウント `davis` 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・アカウント `davis` 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・アカウント `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

項目	説明
<code>/usr/sbin/rmuser</code>	<code>rmuser</code> コマンドが入っています。
<code>/etc/security/passwd</code>	パスワード情報が入っています。
<code>/etc/security/user</code>	ユーザー・アカウントの拡張属性が入っています。
<code>/etc/security/environ</code>	ユーザー・アカウントの環境属性が入っています。
<code>/etc/group</code>	グループの基本属性が入っています。

関連情報:

`chfn` コマンド

`chsh` コマンド

`lsgroup` コマンド

ネットワークの保護

ユーザー、ロール、およびパスワード

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。`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。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、`lssecattr` コマンドまたは `getcmdattr` サブコマンドの項を参照してください。

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

項目	説明
<code>/etc/vfs</code>	仮想ファイルシステムのタイプに関する記述が入っています。

関連情報:

vfs ファイル
 crvfs コマンド
 lsvfs コマンド
 mount コマンド
 ファイルシステム

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。

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・ラインから実行されると、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

項目	説明
<code>-d QueueDeviceName</code>	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。
<code>-q PrinterQueueName</code>	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

項目	説明
/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

smit コマンド

印刷の管理

印刷プーラー

rmwpar コマンド

目的

workload partitionを除去します。

構文

```
/usr/sbin/rmwpar [ -F ] [ -p ] [ -s ] [ -v ] WparName
```

説明

rmwpar コマンドは、指定されたworkload partitionをシステムから削除します。これには、以下のタスクが含まれます。

- workload partitionの構成データをシステムのworkload partitionデータベースから除去する。
- workload partitionのファイルシステムを削除する (-p フラグを指定しない場合)。
- workload partitionのワークロード・マネージャー (WLM) プロファイルを除去する。

-F フラグを指定しない場合、**rmwpar** コマンドは操作のいずれかの部分で最初の障害が起こった時点で停止します。-F フラグを指定した場合は、**rmwpar** コマンドは可能な限り多くの除去対象を除去します。指定されたworkload partitionがアクティブな場合、**rmwpar** コマンドは、-s フラグまたは -F フラグが指定されていない限り失敗します。

フラグ

項目	説明
-F	rmwpar コマンドが大部分の障害をオーバーライドまたは無視する必要があることを指定します。このフラグは、破損した workload partitionsの除去を強制するために使用できます。このフラグは -s フラグを暗黙指定します。

項目	説明
-p	workload partitionに割り当てられている保存除去を除去しません。論理ボリュームまたは既存の論理ボリューム内のサブディレクトリーである構成済みローカル・ファイルシステムは、空にされたり除去されたりすることはありません。このフラグはシステム・workload partitions専用です。このフラグは、rootvg ワークロード・パーティションでは使用できません。このフラグを使用して保存されたファイルシステムは、そのファイルシステムに接続されるworkload partitionを新規に作成するために、次のコマンドで使用できます。
-s	mkwpar -p workload partitionを停止します。このフラグは、 rmwpar コマンドの前に stopwpar コマンドを呼び出すことと同等です。workload partitionのシャットダウンと削除を 1 つのステップで行うには、このフラグを使用します。-F フラグを指定して rmwpar コマンドが実行された場合は、-F フラグを指定して stopwpar コマンドを実行できます。アクティブなworkload partitionに対して -s フラグまたは -F フラグを指定しないで rmwpar コマンドが実行される場合、 rmwpar コマンドは失敗します。
-v	冗長モード。

セキュリティ

アクセス制御: root ユーザーだけがこのコマンドを実行できます。

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権命令を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権についての詳細情報は、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

- 「roy」というworkload partitionを除去するには、次のように入力します。

```
rmwpar roy
```

- 「roy」というworkload partitionを停止して除去し、そのファイルシステム上のデータを保存するには、次のように入力します。

```
rmwpar -p -s roy
```

関連情報:

chwpar コマンド

clogin コマンド

devexports コマンド

lswpar コマンド

mkwpar コマンド

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ーを除去します。

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

項目	説明
-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

System Management Interface Tool

Network Information Service (NIS)

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 [ -b source-address ] [-c config-file] [-k key-file] [-s server] [-p port] [-V] [-y key_id] command
```

説明

rndc コマンドはネーム・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・ライン・オプションまたは引数を指定せずに **rndc** コマンドを実行すると、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 コマンドは 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** コマンドおよび **named** デーモン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鍵を使用します。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 コマンド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鍵を使用すべきかを決めます。

フラグ

項目	説明
-b <i>source-address</i>	サーバーへの接続用の送信元アドレスとして <i>source-address</i> を使用します。IPv4 と IPv6 の送信元アドレスの両方の設定を許可するために、複数インスタンスが許されます。
-c <i>config-file</i>	デフォルトの <code>/etc/rndc.conf</code> ではなく、 <i>config-file</i> 値を構成ファイルとして使用します。
-k <i>key-file</i>	デフォルトの <code>/etc/rndc.key</code> ではなく、 <i>key-file</i> 値を鍵ファイルとして使用します。 <i>config-file</i> の引数が存在しない場合、 <code>/etc/rndc.key</code> 内の鍵を使用して、サーバーに送信されるコマンドを認証します。
-s <i>server</i>	rndc コマンドの構成ファイル内の <code>server</code> 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を指定します。 <i>server</i> 値が無指定の場合、構成ファイルのオプション・ステートメント内の <code>default-server</code> 文節で指定されたホストが使用されます。
-p <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。
-V	詳細ロギングを使用可能にします。
-y <i>keyid</i>	構成ファイルから、 <i>keyid</i> 鍵を使用します。制御メッセージ妥当性検査を成功させるには、 <i>keyid</i> 値は、同じアルゴリズムおよび秘密文字列を使用して <code>named</code> デーモンに知られている必要があります。 <i>keyid</i> 値が無指定の場合、 rndc コマンドは最初に、使用するサーバーの <code>server</code> ステートメント内で <code>key</code> 文節を探し、またはそのホストに <code>server</code> ステートメントがない場合は、 <code>options</code> ステートメントの <code>default-key</code> 文節を探します。 注: 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鍵が含まれます。したがって、この構成ファイルは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保有できません。

rndc コマンド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** コマンド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。

制限

rndc コマンドは `named9` デーモンとだけ働きます。*key_id* 用の共有秘密鍵を、構成ファイルを使用せずに提供することはできません。

関連資料:

8 ページの『named9 デーモン』

241 ページの『nsupdate9 コマンド』

『rndc-confgen コマンド』

関連情報:

dig コマンド

host9 コマンド

rndc-confgen コマンド

```
rndc-confgen [ -a ] [ -b keysize ] [ -c keyfile ] [ -h ] [ -k keyname ] [ -p port ] [ -r randomfile ] [ -s address ] [ -t chrootdir ] [ -u user ]
```

目的

rndc コマンドの構成ファイルを生成します。

構文

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。このコマンドを便利な代替方法として使用して、**rndc.conf** ファイル、対応する制御ステートメント、および鍵ステートメントを **named.conf** に手作業で書き込まなくても済むように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て、**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。

フラグ

項目	説明
-a	自動 rndc 構成を実行します。これは、始動時に rndc コマンドと named デーモンの両方が読み取る /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 コマンドがそれ以上の構成を必要とせずにローカル・ホスト上の named デーモンと通信できるようにする認証キーを定義します。
-b <i>keysize</i>	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 <i>keyfile</i>	-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 コマンドへのオプションと引数の簡易要約を印刷します。
-k <i>keyname</i>	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 <i>port</i>	named デーモンが 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 <i>randomfile</i>	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 またはそれと等価のデバイスを提供しない場合、デフォルトのランダム・データのソースはキーボード入力になります。 <i>randomfile</i> 引数には、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。 keyboard 値は、キーボード入力の使用が必須であることを示します。
-s <i>address</i>	named デーモンが 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。
-t <i>chrootdir</i>	-a フラグと一緒に使用して、 named デーモンが chrooted を実行するディレクトリーを指定します。 rndc.key の追加コピーをこのディレクトリーに関連して書き込んで、それによって、そのコピーが chrooted named により検出されるようにします。
-u <i>user</i>	-a フラグ指定で使用して、生成された rndc.key ファイルの所有者を設定します。 -t フラグも指定されている場合、 chroot 領域にあるファイルだけにその変更済み所有者があります。

例

- 人手による構成を指定せずに **rndc** コマンドを使用するには、以下のコマンドを入力します。

```
rndc-confgen -a
```
- サンプルの **rndc.conf** ファイルを印刷し、対応する制御ステートメントとキー・ステートメントが **named.conf** ファイルに手作業で挿入されるようにするには、以下のコマンドを実行します。

```
rndc-confgen
```

関連資料:

- 8 ページの『named9 デーモン』
- 233 ページの『nslookup コマンド』
- 871 ページの『rndc コマンド』

関連情報:

host9 コマンド

dnssec-keygen コマンド

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、**troff** コマンドのフィルターとして、特に、**-e**、**-h**、**-n**、**-o**、**-r**、**-s**、および **-T** フラグを使用したフィルターとして使用されます。

%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

項目	説明
<i>FormatFlags</i>	nroff コマンド・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。
<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

項目	説明
-m Macro	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。 -m 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 /usr/share/lib/tmac/tmac.bib 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。
-x	%X 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 **html** と一緒に使用したときの例を以下に示します。

関連情報:

addbib コマンド

indxbib コマンド

lookbib コマンド

troff コマンド

rolelist コマンド

目的

ユーザーまたはプロセスのロール情報を表示します。

構文

```
rolelist [-a] [-e | -u username | -p PID]
```

説明

rolelist コマンドは、現行のロールまたは割り当てられているロールについて、ロールおよび権限の情報を呼び出し側に提供します。フラグまたは引数が指定されない場合、**rolelist** コマンドは、実ユーザー ID の呼び出し側に割り当てられているロールのリストと、各ロールのテキスト説明 (ロール・データベースに説明が用意されている場合) を表示します。**-e** フラグを指定すると、セッションで現在有効なアクティブ・ロール・セットに関する情報を出力します。呼び出し側が現在ロール・セッションにない場合に **-e** フラグを指定しても、出力は表示されません。**-a** フラグを指定すると、テキスト記述の代わりに、ロールに関連付けられている権限が表示されます。

また、**rolelist** コマンドでは、特権ユーザーが別のユーザーまたはプロセスに関するロール情報をリストすることもできます。**-u** フラグと一緒にユーザー名を指定すると、特権ユーザーは別のユーザーに割り当てられているロールをリストできます。ユーザーは複数のアクティブ・ロール・セッションを持っている場合があるため、ユーザーのアクティブ・ロール・セットを決めることはできません。したがって、**-u** フラグを指定した場合は、**-e** フラグは指定できません。**-p** フラグと一緒にプロセス ID を指定すると、特権ユーザーはプロセスに関連付けられたロールを表示できます。非特権ユーザーが **-u** または **-p** フラグを指定してこのコマンドを呼び出した場合、コマンドは即時に失敗します。

rolelist コマンドが表示する権限情報は、カーネル・セキュリティ・テーブルから取り出されます。カーネル・セキュリティ・テーブルが更新された後でロール・データベースが変更された場合、取り出される情報はロール・データベースの現在の状態と異なる場合があります。

フラグ

項目	説明
-a	ロール記述の代わりに、各ロールに割り当てられている権限を表示します。
-e	セッションの有効なアクティブ・ロール・セットに関する情報を表示します。
-u <i>username</i>	指定されたユーザーのロール情報を表示します。
-p <i>PID</i>	指定されたプロセスのロール情報を表示します。

セキュリティ

すべてのユーザーが **rolelist** コマンドを実行できます。別のユーザーまたはプロセスのロール情報を照会するためには、以下の権限が必要です。

項目	説明
<code>aix.security.role.list</code>	別のユーザーを対象にこのコマンドを起動するために必要です。
<code>aix.security.proc.role.list</code>	プロセスに関連付けられたロールをリストするために必要です。

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティー」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、`lssecattr` コマンドまたは `getcmdattr` サブコマンドの項を参照してください。

アクセスされるファイル

ファイル	モード
<code>/etc/security/user.roles</code>	r
<code>/etc/security/roles</code>	r

例

1. ユーザー自身に割り当てられているロールのリストを表示するには、次のコマンドを入力します。

```
rolelist
```

次の例のような情報が表示されます。

```
UserAdmin      User Administrator
RoleAdmin      Role Administrator
FSAdmin        File System Administrator
```

2. 割り当てられたロールに関連付けられた権限を表示するには、次のコマンドを入力します。

```
rolelist -a
```

次の例のような情報が表示されます。

```
UserAdmin      aix.security.user
RoleAdmin      aix.security.role
FSAdmin        aix.security.fs
```

3. 特権ユーザーとして、特定のユーザーに割り当てられているロールを表示するには、次のコマンドを入力します。

```
rolelist -u user1
```

次の例のような情報が表示されます。

```
SysInfo        System Information Retrieval
```

関連情報:

`mkrole` コマンド
`ckauth` コマンド
`chuser` コマンド
`swrole` コマンド
RBAC コマンド

roleqry コマンド

目的

一定期間にわたってロールの使用情報を照会します。

構文

```
roleqry {-c [-s ] | -q [ -F <trailListfile> ] [-t <time_period_in_days> ] } user
```

説明

roleqry コマンドは、指定された時間枠にわたってユーザーが使用するロール情報を照会します。

-c フラグを指定すると、そのユーザーはロール情報と許可情報の監査に対応して構成されます。**rbacqry** クラスが、許可とロールの監査に対するイベントを指定して、`/etc/security/audit/config` ファイルに追加されます。ユーザーが既に監査状態にある場合 (ユーザー・エントリーが構成ファイルに存在)、**rbacqry** クラスがそのユーザーに追加されます。そうでない場合、**rbacqry** クラス・パラメーターを指定してユーザー名が `/etc/security/audit/config` に追加されます。**-s** フラグを指定した場合、そのユーザーは監査に対応します。監査サブシステムが既にオンになっている場合、そのサブシステムは再始動されます。監査サブシステムが既にオフになっている場合、その監査サブシステムは始動します。

-q フラグを指定すると、ロール情報用に監査データが照会されます。**-t** フラグを指定すると、その日付から現行システム日付までのロール使用状況が照会され、入手されます。**-t** フラグが無指定の場合、監査がそのユーザーに対して使用可能になっていた期間のロール使用状況が入手されます。このコマンドは、その時間枠の間に使用されたロールの全体セットを表示します。

注: **roleqry** コマンドは、AIX での監査フィーチャーを使用します。**roleqry** コマンドが期待したとおりに機能するには、監査処理をオンにし、ユーザー・セットアップに対応して構成を監査し、指定された時間枠の間に収集された監査データを監査する必要があります。

フラグ

項目	説明
-c	このフラグを使用して、ロール使用状況を監査するためのユーザーを構成します。
-s	このフラグを使用して、監査サブシステムがオフ状態の場合にこのサブシステムを開始します。監査サブシステムが既にオンになっている場合は、そのサブシステムのシャットダウンと再始動を行います。
-q	このフラグを使用して、一定期間を通じてロール使用状況の監査データを照会します。
-F	このフラグを使用して、 <code>trailListFile</code> から監査情報を入手するために監査証跡の名前を読み取ります。監査証跡ファイルの名前は、テキスト行あたり 1 つの名前である必要があります。 -F フラグが指定されていない場合、監査情報の入手元のファイルとしてそのシステムの <code>audit/trail</code> ファイルがデフォルトで使用されます。
-t	このフラグを使用して、許可使用を得るために現在日付からの日数を指定します。

終了状況

項目	説明
0	正常終了。
>0	エラーが発生しました。

セキュリティー

アクセス制御: このコマンドは、root ユーザーに実行 (x) アクセス権限を付与します。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティー」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

ファイル:

- /etc/security/roles
- audit/trail

例

1. Bob が使用するロールを照会するには、以下のコマンドを実行します。

```
roleqry -q Bob
```

2. 過去 20 日間に Simon が使用したロールを照会するには、以下のコマンドを実行します。

```
roleqry -q -t 20 Simon
```

関連情報:

audit コマンド

authqry コマンド

イベント・ファイル

監査の概要

監査の設定

rolerpt コマンド

目的

ロールのセキュリティー機能をレポートします。

構文

```
rolerpt [-R <load_module>] [-C ] [-c | -f ] { "ALL" | role1, role2, .... | -a }
```

```
rolerpt [-R <load_module> ] [-C ] [ -u ] { "ALL" | role1, role2, ... }
```

説明

rolerpt コマンドを使用して、ロールの機能情報 (特権コマンド、特権ファイル、およびユーザー情報など) をレポートします。

-c、**-f**、または **-u** フラグのいずれかを指定できます。**-c** フラグを指定すると、ロールの効力により実行可能な、/etc/security/privcmds データベースに存在する特権コマンドがリストされます。**-f** フラグを指定すると、ロールに割り当てられたユーザーによりアクセス可能な、/etc/security/privfiles データベースに存在する特権ファイルのリストが表示されます。

-u フラグを指定すると、/etc/nscontrol.conf データベースで構成されているロード可能な認証モジュール (LAM) に基づいて、ロールを保有するユーザーのリストが表示されます。**-u** フラグを使用可能なのは、root ユーザーまたは **rolerpt** コマンドに対して許可された特権ユーザーだけです。root ユーザーまたは **aix.security.role.list** 権限を持つユーザーだけが、それらが保持しないロールの機能を表示するレポートを見ることができます。

フラグを何も指定しないと、そのロールに対するすべての機能情報 (コマンド、特権ファイル、およびユーザー情報など) が表示されます。

-a フラグは、アクティブ・ロールの機能を指定します。-u フラグは、-a フラグと一緒に使用できません。root ユーザーまたは権限のあるユーザーは **ALL** キーワードを指定して、システム上のすべてのロールの機能を表示できます。

rolerpt コマンドは、アクティブ・ロールを指定する **-a** フラグ、**ALL** キーワード、またはロール名のコマンド区切りリストなどの入力を受け入れます。ロール名を何も指定しないと、すべての機能情報 (コマンド、特権ファイル、および起動側のロールに関連したユーザー情報など) が表示されます。

フラグ

項目	説明
-a	アクティブ・ロールの機能のみに関するレポートを入手したい旨を指定します。
-c	ロールにより実行可能な特権コマンドのレポートを入手したい旨を指定します。
-C	コロンで区切られたレコードでロール属性を表示します。以下にその例を示します。 #role:attribute1:attribute2: ... role1:value1:value2: ... role2:value1:value2: ...
-f	ロールにアクセス可能な特権ファイル情報のレポートを入手したい旨を指定します。
-R	ロール機能のレポートを入手するロード可能モジュールを指定します。
-u	ロールに割り当てられた、権限のあるユーザー情報のレポートを入手したい旨を指定します。

終了状況

項目	説明
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: このコマンドは、すべてのユーザーに実行 (x) のアクセス権限を付与する必要があります。-u フラグを使用できるのは、root ユーザーか、あるいは **aix.security.role.list** 権限または **aix.security.user.list** 権限を持つユーザーです。root ユーザーまたは **aix.security.role.list** 権限を持つユーザーだけが、**ALL** キーワードを指定でき、それらが保持しないロール機能のレポートを表示できます。

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行します。特権操作を実行できるのは特権ユーザーのみです。権限および特権についての詳細情報は、『特権コマンド・データベース』トピックを検討してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドを確認してください。

ファイル

- /etc/security/roles
- /etc/security/authorizations
- /etc/security/privcmds
- /etc/security/privfiles

例

1. ロール **ManageAllUsers** に関連したコマンドをレポートするには、次のコマンドを実行してください。

```
rolerpt -c ManageAllUsers
```
2. アクティブ・ロールの機能 (すなわち、許可、コマンド、および特権ファイルの情報) をレポートするには、次のコマンドを実行します。

```
rolerpt -a
```

3. コロンで区切られた形式でロール `ManageAllUsers` のすべての機能をレポートするには、次のコマンドを実行してください。

```
rolerpt -C ManageAllUsers
Information similar to the following appears:
#role:commands:privfiles:users
ManageAllUsers:/usr/bin/luser,/usr/bin/mkuser:/var/adm/sulog:Bob,Simon
```

関連情報:

lssecattr コマンド

authrpt コマンド

usrprt コマンド

getcmdattr コマンド

特権コマンド・データベース

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

外部スナップショットへのロールバック

```
rollback [-s ] [ -v ] [-c] snappedFS snapshotObject
```

内部スナップショットへのロールバック

```
rollback [ -v ] -n snapshotName snappedFS
```

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** コマンドの実行前にアンマウントする必要があります。このコマンドの実行中はアクセス不能にされます。指定されたスナップショット (外部スナップショットの場合は *snapshotObject*、内部スナップショットの場合は *snapshotName*) より後で取られたスナップショットは、すべて除去されます。外部スナップショットの場合は、関連付けられた論理ボリュームも除去されます。

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、初期コマンドと同じ *snapshotObject* または *snapshotName* をターゲットにする必要があります。

フラグ

項目	説明
-c	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、 rollback は続行されます。 -c フラグを指定しない場合は、エラー・メッセージが出され、ロールバックは停止します。その場合は fsck コマンドを実行してください。
-n snapshotName	ロールバックに使用する内部スナップショットの名前を指定します。
-s	このフラグが指定された場合は、 rollback 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。
-v	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

項目	説明
<i>snappedFS</i>	ロールバックする JFS2 システム。
<i>snapshotObject</i>	外部スナップショットが戻される元の論理ボリューム。

例

/home/janet/sb ファイルシステムを論理ボリューム **/dev/snapsb** 上の外部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

項目	説明
/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 ] [- i] [-@ WparName]
```

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

項目	説明
-f	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。
-i	ワークロード・パーティション (WPAR) に対してワークロード・パーティション固有の経路指定を使用可能にします。デフォルトで、WPAR からの出力ネットワーク・トラフィックは、グローバル環境から送信されているかのように経路指定されます。 <ul style="list-style-type: none">同じグローバル・システムでホストされているアドレス間のトラフィックは、ループバック・インターフェースを通じて送信されます。グローバル・システム内で構成された、デフォルトの経路を含む経路指定テーブル・エントリーは、ワークロード・パーティションのトラフィックを伝送するために使用されます。 -i フラグを指定して、WPAR 固有の経路指定を使用可能にすると、WPAR は出力トラフィック用に独自の経路指定テーブルを作成して使用します。経路指定エントリーは、ブロードキャスト、ループバック、およびサブネット経路に合わせて、WPAR のネットワーク・アドレスのそれぞれについて自動的に作成されます。
-n	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。
-q	抑止モードを指定し、すべての出力を抑制します。
-C	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、 ioctl 呼び出しのための優先を指定します。
-v	詳細モードを指定し、追加の詳細を印刷します。

項目	説明
-net	<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。
-netmask	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。
-host	<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。
-prefixlen <i>n</i>	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。
-@ <i>WparName</i>	WPAR に関連付けられているネットワーク統計情報 (すなわち <i>@WparName</i> フラグ) を表示します。 <i>@WparName</i> フラグを指定しないと、すべての WPAR のネットワーク統計情報が表示されます。

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

項目	説明
引数	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。
-active_dgd	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。
-cloning	新しい経路のクローンを複製します。
-genmask	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。
-interface	インターフェースの経路指定エントリーを操作します。
-rtt <i>n</i>	往復時間を指定します。
-rttvar <i>n</i>	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。

項目	説明
	<p>-expire <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。</p> <p>-ssthresh <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。</p> <p>-lock 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。-lock 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。</p> <p>-lockrest 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。</p> <p>-if <i>ifname</i> この経路と関連付けるインターフェース (<i>en0</i>, <i>tr0</i> ...) 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。</p> <p>-xresolve 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)。</p> <p>-iface 宛先が直接到達可能であることを示します。</p> <p>-static 手動で追加した経路を指定します。</p> <p>-nostatic 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。</p> <p>-reject 一致した場合、未到達の ICMP を出力します。</p> <p>-blackhole 更新中にパケットを通知なしで廃棄します。</p> <p>-proto1 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。</p> <p>-proto2 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。</p>
コマンド	<p>次の 6 つの選択肢の 1 つを指定します。</p> <p>追加 経路を追加します。</p> <p>flush or -f すべての経路を除去します。</p> <p>delete 特定の経路を削除します。</p> <p>変更 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。</p> <p>モニター 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。</p> <p>get 宛先の経路指定を検索し、表示します。</p> <p>set 経路のポリシーおよび重み属性を設定します。</p>
Family	<p>アドレス・ファミリーを指定します。-inet アドレス・ファミリーがデフォルトです。 -inet6 ファミリーは、すべての後続アドレスが inet6 ファミリーに入っていることを指定します。</p>
Destination	<p>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。<i>Destination</i> 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。</p>
Gateway	<p>パケットのアドレス先のゲートウェイを指定します。<i>Gateway</i> 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。</p>

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストに

については、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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
```

関連資料:

44 ページの『**netstat** コマンド』

関連情報:

gethostbyname コマンド

getnetbyname コマンド

/etc/hosts コマンド

TCP/IP アドレッシング

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・ラインから **routed** デーモンを制御するには、**SRC** コマンドを使用します。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、またはルーティング情報プロトコル

(RIP) しか適用しない **routed** デーモンを使用します。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。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` ファイルを使用します。

`/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す。

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` ファイルの存在を検査します。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・コンピューターの 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・ラインに **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。

項目	説明
startsrc	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。
stopsrc	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。
tracesoff	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。
lssrc	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。

項目	説明
SIGINT	routed 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。
SIGHUP, SIGTERM, または SIGQUIT	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 SIGHUP 、 SIGTERM 、 SIGQUIT のいずれかのシグナルが送信されると、 routed デーモンは終了します。
SIGUSR1	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。
SIGUSR2	パケット・トレースをオフにします。

フラグ

項目	説明
-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 <i>LogFile</i> 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。 Use the html

関連資料:

881 ページの『route コマンド』

関連情報:

gated コマンド

TCP/IP 経路指定

TCP/IP デーモン

TCP/IP プロトコル

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 **umask** の仕様を読み取ります。ログインの記録は、**exportfs** コマンドおよび **enq** コマンドに追加されます。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
\$FILE	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。
\$USER	クライアントへログインしたユーザーの名前。
\$HOST	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

項目	説明
<code>/etc/pcnfsd.conf</code>	<code>rpc.pcnfsd</code> デーモン構成ファイルが入っています。
<code>/var/spool/pcnfs</code>	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

ネットワーク・ファイルシステム (NFS) の概要

印刷の管理

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。

項目	説明
proto.h	ヘッダー・ファイル
proto_xdr.c	XDR ルーチン
proto_svc.c	サーバー側スタブ
proto_clnt.c	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。

項目	説明
RPC_HDR	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。
RPC_XDR	XDR ルーチン内にコンパイルされる際に定義されます。
RPC_SVC	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。
RPC_CLNT	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前で、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

項目	説明
-c	XDR ルーチンにコンパイルします。
-h	C データ定義 (ヘッダー・ファイル) 内にコンパイルします。
-l	クライアント側スタブ内にコンパイルします。
-m	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。
-oOutputFile	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・ファイルシステム (NFS) の概要

プログラミングのためのリモート・プロシージャ・コール (RPC) の概要

NFS コマンドのリスト

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 [ -a -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ます)。

フラグ

項目	説明
-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ジャー 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 デーモンに接続するとき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。
-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、各プロシージャーが要求され、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 アクティビティをモニターするときに使用します。
-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 <i>Portnum</i> 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。
-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ジャー 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。
-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。

例

- 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
```

- 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。

- 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。

- 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。

- 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

項目	説明
<code>/etc/services</code>	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。

関連資料:

428 ページの『`portmap` デーモン』

関連情報:

ネットワーク・ファイルシステム (NFS) の概要

`/etc/services` ファイル

rpvstat コマンド

rpvstat コマンド・マニュアル・ページには、**rpvstat** コマンドの参照情報が記載されています。

目的

RPV クライアント統計情報を表示します。

構文

```
rpvstat -h
```

```
rpvstat [-n] [-t] [-i Interval [-c Count] [-d]] [rpvclient_name . . .]
```

```
rpvstat -N [-t] [-I Interval [-c Count] [-d]]
```

```
rpvstat -m [-n] [-t] [rpvclient_name . . .]
```

```
rpvstat -R [-r][rpvclient_name . . .]
```

```
rpvstat -r [-R] [rpv-device(s)...]
```

```
rpvstat -A [-t] [-i Interval [-d] [-c Count] ] [rpv-device(s)...] |
```

```
rpvstat -C [-t] [-i Interval [-d] [-c Count] ] [rpv-device(s)...]
```

説明

rpvstat コマンドは、以下のような、RPV クライアント・デバイスから入手可能な統計情報を表示します。

- RPV クライアント名
- 接続状況

- 完了した読み取りの合計数
- 読み取りの合計 KB 数
- 読み取りエラーの合計数
- 保留中の読み取りの合計数
- 保留中の読み取りの合計 KB 数
- 完了した書き込みの合計数
- 書き込まれた合計 KB 数
- 書き込みエラーの合計数
- 保留中の書き込みの合計数
- 保留中の書き込みの合計 KB 数
- 非同期入出力の統計情報
- 非同期入出力キャッシュの統計情報

読み取りエラーと書き込みエラーはまとめて表示されます。カウンターに示される数は、アプリケーションに返された入出力エラーの数です。

rpvstat コマンドは、オプションで入出力関連の統計情報をネットワークごとに表示できます。このコマンドでネットワーク要約オプションを使用すれば、以下の追加情報を表示できます。

- ネットワーク・スループット (キロバイト/秒)。このスループットは、モニター・モード中に、ユーザーが指定した間隔回数ごとに計算されます。

rpvstat コマンドは、保留中の統計情報の最高記録値も表示できます。このような最高水準点の履歴の数値には、次の数値があります。

- ネットワークごとの最大保留中読み取り数
- ネットワークごとの最大保留中読み取りキロバイト数
- ネットワークごとの最大保留中書き込み数
- ネットワークごとの最大保留中書き込みキロバイト数

このような統計情報は、次の追加統計情報とともに別の表示で報告されます。

- 入出力操作 (読み取りと書き込みの両方) の再試行回数。このカウントでは、このネットワークまたはデバイスで行われた入出力再試行回数が記録されます。このカウントは、限界に近いネットワークや、障害のあるネットワークに対する指標として使用できます。

また、非同期ミラーリングの統計情報も表示できます。**rpvstat** コマンドで **-A** オプションを使用した場合、全体的な非同期統計情報が出力されます。デバイスごとに統計情報を表示するには、デバイスのリストを指定する必要があります。**-C** オプションを使用すれば、非同期入出力キャッシュ情報を表示できます。

表 1. フラグ

フラグ	説明
-h	コマンドの構文と使用方法を表示します。
-R	RPV クライアントのカウンターをリセットします (root 特権が必要です)。
-t	表示に日時を含めます。
-n	個々のミラーリング・ネットワークの統計情報を表示します。
-N	ミラーリング・ネットワークごとの要約統計情報 (ネットワークごとのスループット速度など) を表示します。
-i Interval	<Interval> 秒ごとに状況を自動的に再表示します。 <Interval> パラメーターの値は、1 以上 3600 以下の整数でなければなりません。 <Interval> パラメーターが指定されていない場合、状況情報は一度のみ表示されます。
-c Count	指示された間隔の <Count> 回数で情報を再表示します。 <Count> パラメーターの値は、1 以上 999999 以下の整数でなければなりません。 <Interval> パラメーターが指定されて、<Count> パラメーターが指定されない場合、いつまでも再表示が行われます。
-m	履歴最大保留値 (最高水準点の値) および累積再試行カウントを表示します。
-d	モニターされた利用可能な統計情報を前回の値からの差分量として表示します。
-A	非同期入出力の統計情報を表示します。
-C	非同期入出力キャッシュの統計情報を表示します。
-r	非同期入出力キャッシュ情報のカウンターをリセットします。 -R および -r オプションと一緒に指定すれば、すべてのカウンターをリセットできます。 root アクセス権限が必要です。

- モニター・モード (-i) では、-d オプションも指定されている場合、統計情報の一部 (完了した読み取り、完了した書き込み、完了した読み取りキロバイト、完了した書き込みキロバイト、およびエラー) が、前回の表示値からの差分量として示されます。このような統計情報の前には、2 回目以降の表示で正符号 (+) が付けられます。差分値は、前回の反復でエラーが検出されたときや、反復の間に構成が変更されたときなど、特定の状況では表示されません。
- RPV クライアント・デバイスのリストがコマンド行に明示的にリストされない場合、使用可能なすべての RPV クライアントのリストがコマンド開始時に作成されます。 モニター・モードでは、この RPV クライアント表示リストは、各表示ループで最新表示されません。つまり、追加または削除された他の RPV クライアントは、コマンドが再度開始されるまで認識されません。
- -i で指定される間隔は、モニター・モードで RPV 統計情報を収集し始めてから表示するまでの 1 回ごとの時間 (秒) です。この間隔は、一連の更新表示と更新表示の間の経過時間を正確に計測した値ではありません。 rpvstat コマンドは、システム・サービスを呼び出すことで、表示する情報の一部を取得しますが、そのサービスが処理を完了するまでの時間を計算に入れません。RPV の数が増えるほど、rpvstat コマンドが情報を収集するのに時間がかかり、モニター・モードでの一連の表示間隔が長くなり、-i で指定された表示間隔よりも時間間隔が大幅に広がる可能性があります。

表 2. オペランド

フィールド	値
rpvclient_name	情報を表示する 1 つ以上の RPV クライアントの名前。RPV クライアントの名前が指定されない場合、すべての RPV クライアントの情報が表示されます。

-A オプションを使用した場合、1 つ以上の非同期デバイスに関する以下の統計情報が出力されます。

- 非同期デバイス名
- 非同期状況: この状況は 1 文字で出力されます。
 - A - デバイスは非同期入出力用に完全に構成されていて、非同期入出力を受け入れることができます。
 - I - 非同期構成が不完全です。

- U - デバイスは非同期構成で構成されていません。従って、同期デバイスとして動作しています。すべての統計情報が 0 として出力されます。
- X - デバイス状況を取得できません。残りのすべての統計情報が 0 として出力されます。
- 完了した非同期リモート書き込みの合計数。書き込みはミラーリングされて完了しています。
- 完了した非同期リモート書き込みの合計キロバイト数。書き込みはミラーリングされて完了していません。
- ミラーリングする保留中の非同期書き込みの合計数。書き込みはキャッシュに格納されています。この書き込みは、LVM に関しては完了していますが、まだミラーリングされていません。
- ミラーリングする保留中の非同期書き込みの合計キロバイト数。書き込みはキャッシュに格納されています。この書き込みは、LVM に関しては完了していますが、まだミラーリングされていません。
- 応答が保留中の書き込みの合計数。この書き込みは保留キューに入っていて、まだキャッシュに書き込まれていません。
- 応答が保留中の非同期書き込みの合計キロバイト数。この書き込みは保留キューに入っていて、まだキャッシュに書き込まれていません。

-C オプションを使用した場合、非同期入出力キャッシュに関する以下の統計情報が出力されます。VG 名が ODM から抽出されます。

- ボリューム・グループ名
- 非同期状況: この状況は 1 文字で出力されます。
 - A - デバイスは非同期入出力用に完全に構成されていて、非同期入出力を受け入れることができます。
 - I - 非同期構成が不完全です。
 - U - デバイスは非同期構成で構成されていません。従って、同期デバイスとして動作しています。すべての統計情報が 0 として出力されます。
 - X - デバイス状況を取得できません。残りのすべての統計情報が 0 として出力されます。
- 非同期書き込み操作の合計
- 最大キャッシュ使用率 (%)
- キャッシュが最高水準点に達した後にキャッシュ・フラッシュを待機していた保留非同期書き込みの数。
- キャッシュが最高水準点の限界に達した後にキャッシュ・フラッシュを待機していた書き込みの割合。
- キャッシュが最高水準点に達した後の最大待機時間 (秒)。
- キャッシュ内の現在のフリー・スペース (キロバイト)

注

- 読み取りおよび書き込みのカウントはバッファごとに集計されます。つまり、アプリケーションの入出力においてバッファのベクトルが単一の読み取り/書き込み呼び出しで渡される場合、その読み取り/書き込みは単一の入出力としてカウントされるのではなく、ベクトル内のバッファの数としてカウントされます。
- 完了した入出力キロバイト数または保留中の入出力キロバイト数は丸められます。KB の少数部分は出力表示では除去されます。
- 表示出力の cx フィールドには、接続状況が表示されます。このフィールドは以下ようになります。

表 3. cx 出力

フィールド	説明
数字	この数字は、RPV クライアントとその RPV サーバーの間にあるアクティブなネットワーク接続の数です。
Y	IP アドレスで表された接続が使用可能であり、機能していることを示します。
N	IP アドレスで表された接続が使用不可であることを示します。
X	必要な情報をデバイス・ドライバから取得できなかったことを示します。その理由として、デバイス・ドライバがロードされていない、デバイスが使用可能状態にない、デバイスが削除された、などがあります。

終了状況

このコマンドは、以下の終了値を戻します。

表 4. 終了状況

フィールド	説明
0	エラーはありません。
>0	エラーが発生しました。

例

- すべての RPV クライアントに関する統計情報を表示するには、次のように入力します。
`rpvstat`
- RPV クライアント `hdisk14` に関する統計情報を表示するには、次のように入力します。
`rpvstat hdisk14`
- RPV クライアント `hdisk23` の統計カウンターをリセットするには、次のように入力します。
`rpvstat -R hdisk23`
- RPV クライアント `hdisk14` の統計情報を表示し、表示を 30 秒ごとに 12 回繰り返すには、次のように入力します。
`rpvstat hdisk14 -i 30 -c 12`
- すべての RPV クライアントの統計情報を表示し、ミラーリング・ネットワークごとの詳細情報を含めるには、次のように入力します。
`rpvstat -n`
- すべてのミラーリング・ネットワークに関する統計情報を表示するには、次のように入力します。
`rpvstat -N`
- すべての RPV クライアントの最大保留値に関する統計情報を表示するには、次のように入力します。
`rpvstat -m`

ファイル

`rpvstat` コマンドは `/usr/sbin/rpvstat` に含まれています。

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

項目	説明
-b Number	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、 rrestore 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。Number 変数の値が大きいくほど、テープ・デバイスからの物理転送量が多くなります。
-f Machine:Device	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、Device 変数にファイル名 (/dev/rmt0 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詳しくは、 rmt スペシャル・ファイルを参照してください。
-h	File 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、 -r フラグまたは -R フラグが指定されている場合は無視されます。

項目

-i

説明

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。

-r

ファイルシステム全体を復元します。

重要: 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、**mkfs** 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、**mkfs** 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、**fsck** コマンドを実行してください。

-R

rrestore 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。**-R** フラグを使用すると、**rrestore** コマンドを中断、再開することができます。

-s *Number*

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。

-t

バックアップを作成したファイルの目次を表示します。**rrestore** 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 (*/*) ディレクトリーに対する相対名です。ただし、ルート (*/*) ディレクトリーのみは例外です。

項目	説明
-v	復元が進行するにつれて、その進行状況を報告します。
-x	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、 -t 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。
-y	rrestore 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。 rrestore 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。
-?	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。

項目	説明
0	正常終了。
>0	エラーが発生しました。

セキュリティ

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**Issecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

例

- 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。

- 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。

- 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。

- 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
```

- 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

項目	説明
/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:

rmt スペシャル・ファイル

mkfs コマンド

fsck システム

ディレクトリー・コマンド

ファイル・コマンド

Rsh コマンド

目的

Bourne シェルの制限付きバージョンを開始します。

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、**Bourne** シェルの制限付きバージョンを開始します。これは、より制御されたシェル環境を必要とするインストールに役立ちます。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。

フラグ

Bourne シェルは、シェルがコマンド・ラインで開始された場合にのみ、以下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・ラインからコマンド・ファイルへ引き渡されます。

項目	説明
-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がない場合、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 <i>String</i>	<i>String</i> 変数から読み取ったコマンドを実行します。 <i>String</i> 変数の値から特定パラメーター 0 の値を設定し、残りの <i>Parameter</i> 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。 <ul style="list-style-type: none"> • 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ではない。
-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 <i>timeout</i>	サーバーからの応答がない場合、 <i>timeout</i> 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

項目	説明
/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

Bourne シェル shell

Bourne シェル組み込みコマンド

Bourne シェルにおける変数置換

/etc/passwd ファイル

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 [ -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・ラインからリモート・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・ライン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・システムでは、この方法は不要。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。詳細については、`kvalid_user` 関数を参照してください。

リモート・コマンドの実行

リモート・コマンドの実行時に、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
```

このリンクが設定された後、`HostName` パラメーター、およびコマンド・ラインから `Command` パラメーターで指定したコマンドを指定できます。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` コマンド

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。このような目的では、レルムは DCE セルと同義です。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必要があ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てはならないことを指定します。

-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 AIX 認証のみを使用します。

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。

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、そのファイルはターゲット・システム上のローカル・ユーザーの **\$HOME** ディレクトリーになければなりません。ローカル・ユーザーおよびすべてのユーザーがリストされる必要があります。そうしないと、**rsh** コマンドに許可されているサービスが考慮されません。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** ファイル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。アカウントの所有者は、**.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、**.klogin** ファイルへの読み取りおよび書き込みアクセス (許可 = 600) は AIX 所有者 ID だけが持つようにする必要があります。

RBAC ユーザーおよび Trusted AIX ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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・ファイルをリモート・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 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

項目	説明
\$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 (適用できる場合) 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 (適用できる場合) 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 AIX 並列システム・サポート・プログラム (PSSP) 管理ガイド」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 でアクセスできます。

Kerberos に関する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 でアクセスできます。

関連資料:

648 ページの『`rcp` コマンド』

関連情報:

`ftp` コマンド

通信およびネットワーク

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` デーモンによって始動されます。また、`SRC` コマンドを使用してコマンド・ライン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 System Resource Controller (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・ラインで `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

`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・ライン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 **chdir** サブルーチン内でローカル・ユーザー名を検索します。ファイル内の調査またはディレクトリ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ます。
 - d. **\$HOME/.rhosts** の認証が正しく行われない場合、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、初期接続に 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・ラインを渡します。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・ラインで **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。

項目	説明
startsrc	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。
stopsrc	サブシステム、サブシステムのグループ、またはサブサーバーを終了します。
lssrc	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

項目	説明
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、 <i>.profile</i> 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、 <i>.profile</i> 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* ファイルの **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** 属性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* ファイル内の **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* ファイル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:

lssrc コマンド

/etc/inetd.conf コマンド

/etc/services コマンド

TCP/IP デーモン

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

項目	説明
/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/services	インターネットで利用できる各サーバーのエントリーが入っています。

関連資料:

82 ページの『nfsstat コマンド』

関連情報:

inetd コマンド

ネットワーク・ファイルシステム (NFS) の概要

NFS コマンドのリスト

rsyslogd デーモン

目的

システム・メッセージを記録します。

説明

rsyslogd デーモンはソケットを読み取り、/etc/rsyslog.conf 構成ファイルで指定されている宛先にメッセージ行を宛先に送信します。**rsyslogd** デーモンは、起動時にこの構成ファイルを読み取ります。ソース・マスターから以下のコマンドを使用して **rsyslogd** デーモンを開始できます。

```
startsrc -s rsyslogd
stopsrc -s rsyslogd
```

startsrc オプションは **rsyslogd** デーモンを開始します。複数の **rsyslogd** デーモンを開始するには、-i コマンド・ライン・オプションを使用して、新しい pid ファイルごとに startsrc オプションを実行します。startsrc -a フラグを使用すると、startsrc コマンドが **rsyslogd** デーモンの引数を指定します。この引数は、シェルによる解釈を避けるために、二重引用符で保護する必要があります。

stopsrc オプションは、**rsyslogd** デーモンのすべてのインスタンスを停止します。特定のインスタンスを停止するには、`-p <pid>` オプションを指定する必要があります。

`stopsrc -p <pid of syslogd daemon>`

デフォルトのロギング・アプリケーション:

rsyslogd デーモンは、インストール後、即座に開始することはできず、**syslogd** デーモンを使用したシステム・メッセージの記録が継続して行われます。**rsyslogd** デーモンによってメッセージがデフォルトで記録されるように構成するには、`-r` オプションを使用して `syslog_ssw` スクリプトを実行してください。

rsyslogd デーモンは、システム・メッセージを記録するように構成された後、デフォルトのコマンド・ライン引数 `-c5` で開始します。このオプションにより、確実に、**rsyslogd** デーモンは通常モードで開始し、旧バージョンとは互換性がないものになります。

デフォルトの `rsyslog.conf` ファイル:

rsyslogd デーモンを構成および使用するには、資料の該当するセクションを参照してください。

インストール後、デフォルトの `/etc/rsyslog.conf` 構成ファイルには以下の情報が入っています。

```
#####  
# Rsyslog is free software: it is distributed under the          #  
# terms of the GNU General Public License as published by      #  
# the Free Software Foundation, under version 3 of the License. #  
#                                                                #  
# if you experience problems, check                            #  
# http://www.rsyslog.com/doc/troubleshoot.html for assistance  #  
#                                                                #  
# Load the UNIX socket for local communication                 #  
#   $ModLoad imuxsock                                          #  
#                                                                #  
# Load the UDP module for remote communication                 #  
#   $ModLoad imudp                                             #  
#                                                                #  
# Run the UDP server on the default port 514                   #  
#   $UDPServerRun 514                                          #  
#                                                                #  
#####
```

pureScale API をサポートする AIX 固有のパラメーターなどを除くと、`syslog.conf` ファイルのほぼすべてのパラメーターは、**rsyslogd** デーモンによって機能します。`syslog.conf` ファイルをサポートされる `rsyslog.conf` ファイルに変換するには、`-c` オプションと一緒にスイッチ・スクリプトを使用する必要があります。

スイッチ・スクリプトの使用法

`syslog_ssw [-r | -s | -c SourceSyslogConffile DestRsyslogConffile]`

項目	ディスクリプター
-r	デフォルトのロギング・アプリケーションとして rsyslog デーモンに切り替えます。
-s	デフォルトのロギング・アプリケーションとして syslog デーモンに切り替えます。
-c	syslog.conf ファイル内の構成規則を rsyslog.conf ファイル内の規則に変換します。ただし、 rsyslogd デーモンによって認識されない AIX 固有のパラメーターは変換時に削除されます。

-r または -s オプションを使用してデフォルトのロギング・アプリケーションを切り替える場合、この選択は再始動の後も持続します。

startsrc -s syslogd コマンドは、設定されているデフォルトのロギング・アプリケーションに基づいて **rsyslogd** または **syslogd** デーモンを起動します。

syslog_ssw スクリプトは、デフォルトでは存在しておらず、**rsyslogd** デーモンのインストール後に使用可能になります。

例

1. 既存の **syslogd** デーモンを停止して、**rsyslogd** デーモンを始動するには、次のコマンドを実行します。

```
syslog_ssw -r
```

2. 既存の **rsyslogd** デーモンを停止して、**syslogd** デーモンを始動するには、次のコマンドを実行します。

```
syslog_ssw -s
```

3. **syslog.conf** ファイルを **rsyslog.conf** ファイルに変換して、**rsyslog.conf** ファイルが存在しない場合にこのファイルを作成するには、次のコマンドを実行します。

```
syslog_ssw -c syslog.conf rsyslog.conf
```

この変換により、AIX 固有のパラメーターが削除され、新たに作成されたファイルを **rsyslogd** デーモンで使用できるようになります。

4. デフォルトのロギング・アプリケーションを開始するには、次のコマンドを実行します。

```
startsrc -s syslogd
```


デフォルトのロギング・アプリケーションは、**syslogd** デーモンまたは **rsyslogd** デーモンの場合があります。

ファイル

項目	ディスクリプター
/etc/rsyslog.conf	rsyslogd デーモンの出力を制御します。
/var/run/rsyslogd.pid	プロセス ID が入っています。

関連情報:

syslogd デーモン

 **rsyslog** マニュアル

rtcd デーモン

目的

ファイル修正イベントをモニターして結果の準拠性違反をチェックし、管理者にアラートを通知します。

説明

rtcd デーモンは、`/etc/security/rtc/rtcd.conf` ファイルに定義されている構成情報を読み取ります。

rtcd デーモンは **aixpert** コマンドを実行して、起動時の準拠性違反がないかチェックします。違反が判別された場合、このデーモンは `/etc/security/rtc/rtcd.conf` ファイルに指定されている受信側に電子メールでアラートを通知します。

rtcd デーモンは、`/etc/security/rtc/rtcd_policy.conf` ファイルに指定されているファイルについてファイルの変更を引き続きモニターします。変更されたファイルがあると、**rtcd** は、**aixpert** コマンドを実行して、準拠性違反がないかをチェックし、違反があれば電子メールでアラートを送信します。

リアルタイム・コンプライアンスが正常に構成されると、**rtcd** デーモンは SRC の制御下に置かれます。

rtcd デーモンは、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管理する必要があります。

セキュリティ

rtcd デーモンは、`root` ユーザーとシステム・グループによって所有されています。`root` ユーザーおよび `aix.system.config.src` 権限を持つユーザーだけが、このコマンドの管理が許可されています。

例

1. **rtcd** デーモンを開始するには、次のコマンドを入力します。

```
# startsrc -s rtcd
```

2. **rtcd** デーモンをチェックするには、次のコマンドを入力します。

```
# lssrc -s rtcd
```

3. **rtcd** デーモンを停止するには、次のコマンドを入力します。

```
# stopsrc -s rtcd
```

ファイル

項目	説明
<code>/etc/security/rtc/rtcd_policy.conf</code>	rtcd デーモンの構成情報が入っています。
<code>/etc/security/rtc/rtcd.conf</code>	<code>root</code> ユーザーに、読み取り (r) および書き込み (w) のアクセス権限を付与します。

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。

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。共用オブジェクトは、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

項目	説明
-F <i>ObjsLibs</i> ...	ObjsLibs を、生成された ld コマンドの先頭に追加します。 ObjsLibs 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 (ld command's -l (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用オブジェクトについて、 ObjsLibs を ld コマンドに追加します。
-l	(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 .imp を持ち、エクスポート・ファイルは接尾部 .exp を持ちます。 File がモジュールである場合は、 rtl_enable 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。 File 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。
-o <i>Name</i>	File.new 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 -R フラグと一緒に使用しないでください。
-R	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 -o 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

項目	説明
ファイル	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。

項目	説明
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

項目	説明
/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:

ld コマンド

共有オブジェクトと実行時リンク

runacct コマンド

目的

日次アカウントティングを実行します。

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ティングのシェル・プロシーチャーです。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ティング・データ・ファイルを処理して、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)/rprtmmdd` 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については、『`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す。

`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
SETUP	アクティブ・アカウント・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。
WTMPFIX	<code>wtmp</code> 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。
CONNECT1	<code>acctcon1</code> 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。
CONNECT2	接続セッション・レコードを、合計アカウント・レコードに変換します (<code>tacct.h</code> フォーマット)。
PROCESS	プロセス・アカウント・レコードを、合計アカウント・レコードに変換します (<code>tacct.h</code> フォーマット)。
MERGE	接続アカウント・レコードとプロセス合計アカウント・レコードを組み合わせます。
FEES	<code>chargefee</code> コマンドの出力を合計アカウント・レコードに変換し (<code>tacct.h</code> 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・レコードと組み合わせます。
DISK	ディスク・アカウント・レコードを、接続、プロセス、および料金合計アカウント・レコードと組み合わせます。
QUEUEACCT	キュー (プリンター) アカウント・レコードをソートして、それらを合計アカウント・レコードに変換し (<code>tacct.h</code> フォーマット)、さらに他の合計アカウント・レコードと組み合わせます。
MERGETACCT	<code>dayacct</code> レポート・ファイル内の日次合計アカウント・レコードを、 <code>/var/adm/acct/sum/tacct</code> レポート・ファイル内の要約合計アカウント・レコードとマージします。
CMS	<code>/var/adm/acct/sum(x)/cms</code> ファイル内でコマンド要約を作成します。
USEREXIT	<code>/var/adm/siteacct</code> 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。
CLEANUP	一時ファイルを削除して終了します。

`runacct` プロシージャの再始動

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トिंगをどの月日に関して再実行するのかを指定します。**runacct** プロシージャーでは、**/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・ラインで希望する **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トिंग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

項目	説明
-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) アクセス権を与えます。

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティ」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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・ラインを **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。

- システム・リソースに対して、コマンド・ライン (**runacct** コマンドを始動させる) から日アカウントリング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** プロシージャ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。

- 特定の日付に対してシステム・アカウントリング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。

- 特定の状態で、特定の日付に対するシステム・アカウントリング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

項目	説明
/var/adm/wtmp	ログイン/ログオフ・ヒストリー・ファイル。
/var/adm/pacct*	プロセス・アカウントリング・ファイル。
/var/adm/acct/nite(x)/dayacct	ディスク使用状況アカウントリング・ファイル。
/var/adm/qacct	アクティブ・キュー・アカウントリング・ファイル。
/var/adm/fee	ユーザーに課される料金のレコード。
/var/adm/acct/sum(x)/*	コマンド・ファイルと合計アカウントリング要約ファイル。
/var/adm/acct/nite(x)/ptacct*.mmd	pacct ファイルの連結バージョン。
/var/adm/acct/nite(x)/active	runacct メッセージ・ファイル。
/var/adm/acct/nite(x)/lock*	runacct コマンドの同時呼び出しを防止します。
/var/adm/acct/nite(x)/lastdate	runacct が実行された最後の日付が入っています。
/var/adm/acct/nite(x)/statefile	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

fwtmp コマンド

utmp、wtmp、failedlogin

アカウントリング・サブシステムの設定

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。

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。

-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。

-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」フォーマット (1 行ごとに 1 つのエントリー) を指定します。これは、デフォルトの表示形式です。

-n node_name

クラス・アクションを実行するノードの名前を指定します。このフラグは、必ず、**-c** フラグと一緒に使用します。

-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。
- ノード名の左の空白文字はすべて無視されます。
- ノード名の右の空白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。

-r "resource_handle"

リソース・ハンドルを指定します。リソース・ハンドルは次のフォーマットで指定する必要があります。

```
"0xnnnn 0xnnnn 0xxxxxxxx 0xxxxxxxx 0xxxxxxxx 0xxxxxxxx"
```

ここで *n* は、16 進数の文字です。このフラグを使用して、リソースに対し、*resource_handle* に一致するアクションを呼び出します。

-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、「*Administering RSCT*」を参照してください。

-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

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・ラインに *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・ライン・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・ラインに間違ったフラグが指定されました。
- 4 コマンド・ライン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・ライン入力に基づく 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 (Resource Monitoring and Control) 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 高信頼性スケーラブル・クラスター・テクノロジー (RSCT) (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.Example** に対して **TestClassAction** リソース・クラス・アクションを呼び出すには、次のように入力します。

```
runact -c IBM.Example 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

/opt/rsct/bin/runact

runact コマンドが入っています。

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

項目	説明
/usr/bin/runcat	runcat コマンドが入っています。

関連情報:

dspscat コマンド
dspsmsg コマンド
gencat コマンド
catopen コマンド
メッセージ機能

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。

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す。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 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・ラインに指定されている **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** が空だった場合は、**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。

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 `lpacl` 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・ライン・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・ラインに間違ったフラグが指定されました。
- 4 コマンド・ライン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・ライン入力に基づく 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 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

/opt/rsct/bin/runlpcmd

runlpcmd コマンドが入っています。

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・ライン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。**inetd** デーモンは必ず使用してください。

注:

1. 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。
2. 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

項目	説明
-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

1. 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
```

2. 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

項目	説明
html	

関連資料:

912 ページの『**rstatd** デーモン』

関連情報:

sort コマンド

NFS コマンドのリスト

inetd コマンド

ネットワーク・ファイルシステム (NFS) の概要

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ー (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

項目	説明
-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

- 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
```

- 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

項目	説明
<code>/var/spool/rwho/whod.*</code>	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。

関連資料:

938 ページの『**rwho** コマンド』

940 ページの『**rwhod** デーモン』

関連情報:

通信およびネットワーク

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使用することにより、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使用することはできません。

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

項目	説明
-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使用して、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使用して、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使用して、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使用して、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使用して、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使用して、すべてのエントリーを表示します。
-r "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 (rlogin 、 rcp 、 rsh 、または rprint) を実行できるかを指定する /etc/hosts.equiv 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。 -r 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。
-s	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、 -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。
-X	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、 -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。
-Z	-s フラグは、 -Z フラグを指定した場合に必要です。 -Z 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。

セキュリティー

RBAC ユーザーおよび **Trusted AIX** ユーザーへの注意: このコマンドは特権操作を実行できます。特権命令を実行できるのは特権ユーザーのみです。権限および特権については、「セキュリティー」の『特権コマンド・データベース』を参照してください。このコマンドに関連した特権および権限のリストについては、**lssecattr** コマンドまたは **getcmdattr** サブコマンドの項を参照してください。

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、**rcp** 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。

関連資料:

909 ページの『**rshd** デーモン』

関連情報:

lpd コマンド

ftpusers コマンド

hosts.equiv コマンド

hosts.lpd コマンド

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

項目	説明
-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

項目	説明
<code>/etc/inetd.conf</code>	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。

関連資料:

938 ページの『`rwho` コマンド』

関連情報:

`who` コマンド

`inetd` コマンド

ネットワーク・ファイルシステム (NFS) の概要

NFS コマンドのリスト

`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

項目	説明
<code>/etc/inetd.conf</code>	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。
<code>/etc/inetd.conf</code>	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。

関連資料:

933 ページの『`rusers` コマンド』

関連情報:

`inetd` コマンド

ネットワーク・ファイルシステム (NFS) の概要

NFS コマンドのリスト

`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 **startpdomain** 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 **startpnode** 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

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

項目 説明

-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

項目	説明
/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・グループについての情報が入っています。

関連資料:

『rwalld デーモン』

関連情報:

wall コマンド

inetd コマンド

ネットワーク・ファイルシステム (NFS) の概要

NFS コマンドのリスト

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

項目	説明
/etc/inetd.conf	TCP/IP 構成を指定します。

関連資料:

937 ページの『rwall コマンド』

関連情報:

inetd コマンド

ネットワーク・ファイルシステム (NFS) の概要

NFS コマンドのリスト

rwho 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

rwho [**-a**]

説明

/usr/bin/rwho コマンドは、**rwhod** 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、**rwho** 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、**rwhod** 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

項目	説明
-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。

```
rwho
```

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

項目	説明
/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。

関連資料:

930 ページの『**ruptime** コマンド』

938 ページの『**rwho** コマンド』

関連情報:

who コマンド

services コマンド

通信およびネットワーク

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・ラインから **rwhod** デーモンを制御するには、**SRC** コマンドを使用します。システム起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 **whod.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・ラインに **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。

項目	説明
stopsrc	サブシステム、サブシステムのグループ、またはサブサーバーを終了します。
traceson	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。
tracesoff	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。
tracesoff	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・ライン上で使用できます。**-s** フラグは、後に続くサブシステムが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る場合は、**trpt** コマンドを使用してください。

ファイル

項目	説明
/etc/utmp	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。
/var/spool/rwho/*	rwho コマンドと ruptime 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。
/var/spool/rwho/whod.HostName	HostName 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。

関連資料:

930 ページの『**ruptime** コマンド』

938 ページの『**rwho** コマンド』

関連情報:

who コマンド

services コマンド

TCP/IP デーモン

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、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、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。

〒103-8510

東京都中央区日本橋箱崎町19番21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

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。

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。

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 Director of Licensing

IBM Corporation

North Castle Drive, MD-NC119

Armonk, NY 10504-1785

US

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。

記載されている性能データとお客様事例は、例として示す目的でのみ提供されています。実際の結果は特定の構成や稼働条件によって異なります。

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にお願います。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

表示されている IBM の価格は IBM が小売り価格として提示しているもので、現行価格であり、通知なしに変更されるものです。卸価格は、異なる場合があります。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

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。これらの名前はすべて架空のものであり、類似する個人や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。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生した創作物には、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年).

このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。

© Copyright IBM Corp. _年を入れる_.

プライバシー・ポリシーに関する考慮事項

サービス・ソリューションとしてのソフトウェアも含めた IBM ソフトウェア製品（「ソフトウェア・オフアリング」）では、製品の使用に関する情報の収集、エンド・ユーザーの使用感の向上、エンド・ユーザーとの対話またはその他の目的のために、Cookie はじめさまざまなテクノロジーを使用することがあります。多くの場合、ソフトウェア・オフアリングにより個人情報が収集されることはありません。IBM の「ソフトウェア・オフアリング」の一部には、個人情報を収集できる機能を持つものがあります。ご使用の「ソフトウェア・オフアリング」が、これらの Cookie およびそれに類するテクノロジーを通じてお客様による個人情報の収集を可能にする場合、以下の具体的事項を確認ください。

この「ソフトウェア・オフアリング」は、Cookie もしくはその他のテクノロジーを使用して個人情報を収集することはありません。

この「ソフトウェア・オフアリング」が Cookie およびさまざまなテクノロジーを使用してエンド・ユーザーから個人を特定できる情報を収集する機能を提供する場合、お客様は、このような情報を収集するにあたって適用される法律、ガイドライン等を遵守する必要があります。これには、エンドユーザーへの通知や同意の要求も含まれますがそれらには限られません。

このような目的での Cookie などの各種テクノロジーの使用について詳しくは、『IBM オンラインでのプライバシー・ステートメントのハイライト』(<http://www.ibm.com/privacy/jp/ja/>)、『IBM オンラインでのプライバシー・ステートメント』(<http://www.ibm.com/privacy/details/jp/ja/>) の『クッキー、ウェブ・ビーコン、その他のテクノロジー』というタイトルのセクション、および『IBM Software Products and Software-as-a-Service Privacy Statement』(<http://www.ibm.com/software/info/product-privacy>) を参照してください。

商標

IBM、IBM ロゴおよび [ibm.com](http://www.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。

INFINIBAND、InfiniBand Trade Association、および INFINIBAND デザイン・マークは、INFINIBAND Trade Association の商標またはサービス・マークです。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Windows は、Microsoft Corporation の米国およびその他の国における商標です。

Java およびすべての Java 関連の商標およびロゴは、Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

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。

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・システム

基本ファイルの作成

`nulladm` コマンドを使用する 276

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 511

実行

`runacct` コマンドを使用する 917

セッション・レコードの表示 449

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 450

ファイルの削除

`remove` コマンドを使用する 700

プリンター・レコード

作成 303

印刷

ジョブの保持 593

スプールされたジョブの移動 595

印刷キュー

新規の作成 389

内のジョブの優先順位付け

`qpri` コマンドを使用する 603

印刷ジョブ・マネージャ

印刷ジョブの処理

`piobe` コマンドを使用する 372

ファイルの印刷

`pioout` コマンドを使用する 393

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 405

インターネット

ドメイン・ネーム・サーバーの照会 233

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 365

オブジェクト・ファイル

プロファイル・データの表示 490

[カ行]

仮想 RAM ディスク

除去

`rmramdisk` コマンドを使用する 832

仮想プリンター 377, 383

仮想プリンター (続き)

除去

`rmvirprt` コマンドを使用する 868

属性値の表示、`lsvirprt` コマンド使用の 370
の属性値の表示 370

仮想メモリー

システムのページ・サイズの表示 309

環境

変数値の印刷 456

疑似デバイス・プリンター 386

キュー

スプールされたジョブの移動 595

スプールされたジョブの保持 593

共用ログイン・ポート 545

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 58

ゲーム

クイズ 615

数値書き込みゲーム 276

コマンド

チューナブル・パラメーター 72

の異なる優先順位の設定

`nice` コマンドを使用する 86

パラメーター・タイプ 72

ハングアップせずに実行 227

`namerslv` 9

`netstat` 44

`newform` 56

`nfso` 72

`nim` 88

`nimconfig` 144

`niminit` 151

`nlssrc` 177

`no` 203

`ntpq` 263

`ntpq4` 268

`oslevel` 298

`passwd` 312

`pic` 356

`pkgtrans` 413

`pr` 446

`preparevsd` 451

`preprnode` 452

`printf` 456

`prs` 508

`ps` 518

`psroff` 549

`qdaemon` 592

`quot` 617

`quota` 619

コマンド (続き)

- quotacheck 621
- quotaoff 623
- quotaon 623
- ras_logger 636
- rc 644
- rcp 648
- rc.powerfail 645
- rdist 656
- read 672
- recfgct 677
- refrsrc 691
- refsensor 693
- removevsd 701
- remsh 905
- rendev 702
- resetrsrc 714
- resumevsd 746
- rev 747
- rmaudrec 766
- rmcctrl 777
- rmcomg 788
- rmcondition 790
- rmcondresp 792
- rmddel 797
- rmdev 798
- rmdir 801
- rmlpcmd 812
- rmm 817
- rmnamsv 819
- rmnfsexp 821
- rmramdisk 832
- rmresponse 833
- rmrpdomain 837
- rmrpnode 840
- rmsensor 850
- rmss 854
- rmvirprt 868
- rsh 905
- runact 921
- runlpcmd 925
- ruser 931
- rvsdrestrict 935

[サ行]

索引

主題ページの索引の作成

ndx コマンドを使用する 22

順列索引の生成

ptx コマンドを使用する 563

サブサーバー

SRC オブジェクト定義の除去 853

サブシステム

通知メソッドの除去 824

サブシステム (続き)

のリフレッシュの要求

refresh コマンドを使用する 690

SRC オブジェクト・クラスからの定義の除去 859

参考文献データベース

印刷

roffbib コマンドを使用する 874

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 860

システム

リブート

reboot コマンドを使用する 675

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 725

ブート・イメージ 725

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 312

システム・テーブル

の内容の解釈 553

システム・リソース・コントローラー 824, 859

始動

標準の始動初期化の実行

rc コマンドを使用する 644

出力、標準

書き込み

pr コマンドを使用する 446

順列索引

生成

ptx コマンドを使用する 563

状況

プロセス、表示

ps コマンドを使用する 518

初期化、始動

rc コマンドを使用する 644

数式テキスト

フォーマット

neqn コマンドを使用する 24

スクリプト

enotifyevent 229

notifyevent 229

制限付きのシェル 903

ソース・ファイル 318

[タ行]

端末

reset コマンドを使用した初期化 713

reset コマンドを使用した特性の設定 713

端末の設定

現行ウィンドウ・サイズへの設定 719

遅延ログイン・ポート 341

着信メッセージ

ユーザーへの通知 655

- 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 203
- 通知オブジェクト・クラス (SRC)
 - 通知メソッド定義の除去 824
- 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 931
- デーモン
 - ndpd-router 17
 - nfsrgyd 81
 - pcnfsd 889
 - pppattachd 432
 - routed 886
 - rshd 909
- ディスプレイ
 - ファイル
 - 画面に合わせたフォーマット 352
- ディレクトリー
 - 削除 761
 - 除去 761
 - のパス名
 - 表示 574
- テキスト
 - のフォーマットの変更
 - newform コマンドを使用する 56
- デバイス構成コマンド
 - restbase 725
 - restbase コマンド 725
- デルタ・ファイル
 - 除去
 - rmdel コマンドを使用する 797
- 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 236
 - nsupdate4 コマンド 237
- ドライバー
 - プリンターのフォーマット
 - pioformat コマンドを使用する 381

[ナ行]

- 内の参考資料の検出
 - refer コマンドを使用する 685
- 内への参考資料の挿入
 - refer コマンドを使用する 685
- ネットワークの CPU 使用量 29
- 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 88
- ネットワーク・インストール・マネージャー 140, 144
- 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 203

[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 492
- パス名
 - 表示 574
- 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 577
 - 変更
 - passwd コマンドを使用する 312
 - ユーザー・パスワードの管理
 - pwdadm コマンドを使用する 575
- 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 405
- パラメーター
 - 名前 702
 - NewName 702
- 非同期接続
 - PPP サブシステム
 - pppdial コマンド 440
- ファイル
 - 圧縮
 - pack コマンドを使用する 304
 - アンパック
 - pcat コマンドを使用する 340
 - 行の番号付け 175
 - 削除 761
 - 指定されたフォーマットでの表示 279
 - 除去 761
 - ディスプレイに合わせたフォーマット 352
 - 内容の表示 352
 - の各行の文字の順序を逆にする
 - rev コマンドを使用する 747
 - パス名の検査
 - pathchk コマンドを使用する 323
 - 標準出力への書き出し
 - pr コマンドを使用する 446
 - 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 726
 - 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 12
- ファイルシステム
 - プロトタイプ・ファイルの構成
 - proto コマンドを使用する 505
 - 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 899
- 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 656
- 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 656

- 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 656
- 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の、rdump コマンドを使用する 670
- 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 56
- フォルダー
 - 削除
 - rmf コマンドを使用する 803
 - に含まれるメッセージの除去
 - rmf コマンドを使用する 803
- フォント
 - コピー
 - piofontin コマンドを使用する 380
- 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 656
- 物理ボリューム 710
- プリンター
 - 上のジョブの開始
 - qprt コマンドを使用する 605
 - 上のジョブの取り消し
 - qcan コマンドを使用する 588
 - 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 395
 - 新規の追加 389
 - 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 613
 - バースト・ページ
 - pioburst コマンド 374
 - フォント
 - コピー 380
- プリンター、仮想
 - 除去
 - rmvirprt コマンドを使用する 868
- プリンター定義ファイル
 - 拡張または縮小 376
- プリンターの定義
 - 仮想プリンターの 377
- プリンターのバックエンド・コマンド
 - piodmng 379
 - piolsvp 383
 - piomgpdev 386
 - piomkapqd 387
 - piomkpq 389
 - piomsg 391
- 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 830
 - rmquedev コマンドを使用する 831
 - の状況の表示
 - qchk コマンドを使用する 590
- 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 587
- プリンター・バックエンド
 - へのメッセージの送信 391
- プリンター・フォーマッター
 - 始動
 - pioformat コマンドを使用する 381
- プロセス
 - の状況の表示
 - ps コマンドを使用する 518
- 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 831
- プロット、HP-GL ファイルの 417, 418
- プロット・ファイル
 - PostScript への変換
 - psplot コマンドを使用する 546
- プロンプト・エディターの始動、MH 503
- ページング・スペース
 - 除去 828
- 別名データベース
 - 作成 55
- ポート
 - ログイン 342
- ホスト
 - の状況の表示
 - ruptime コマンドを使用する 930
 - ローカルのリモートとの接続
 - rlogin コマンドを使用する 757
- 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 656
- ボリューム・グループ
 - 物理ボリュームの除去
 - reducevg コマンドを使用する 683
 - ボリューム・グループの再作成 679
 - 論理ボリュームの除去
 - rmlv コマンドを使用する 814
 - reorgvg コマンドを使用した物理区画割り当ての再編成 705

[マ行]

- マニュアル・ページ
 - rmccli 772
- メールのフォルダーへの取り込み、MH 654
- メール・コマンド
 - newaliases 55
 - rmail 764
- 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 817
 - 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 687
 - 直前のメッセージの表示
 - prev コマンドを使用する 454

- メッセージ (続き)
 - 次のメッセージの表示
 - next コマンドを使用する 63
 - 内容による選択
 - pick コマンドを使用する 363
 - に対する応答
 - repl コマンドを使用する 706
 - バック・ファイルへの保存 653
 - 標準出力への書き出し 61
- メッセージ機能コマンド
 - runcat 925
- メッセージ経路指定 431
- 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 817
- メッセージ・シーケンス
 - 作成
 - pick コマンドを使用する 363
 - 変更
 - pick コマンドを使用する 363
- メッセージ・ソース・ファイル
 - 作成 925
- メッセージ・ソース・ファイルの作成 925
- メモリー管理
 - システムのページ・サイズの表示 309
- 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 747

[ヤ行]

- ユーザー
 - アカウントの除去
 - rmuser コマンドを使用する 865
 - に関するパスワード情報の検証
 - pwdck コマンドを使用する 577
 - のパスワードの管理
 - pwdadm コマンドを使用する 575
 - パスワードの変更
 - passwd コマンドを使用する 312

[ラ行]

- ライブラリー
 - アーカイブ・ライブラリーの変換 629
- 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 230
- リモート・マシンのデバイス
 - ファイルのバックアップ、rdump コマンドを使用する 670
- リンク
 - 除去 761
- ルーター
 - NDP および RIPng デーモン
 - ndpd-router デーモンを使用する 17

- ログイン・ポート
 - 共用
 - 使用可能化 545
 - リスト 545
 - 使用可能化
 - pdelay コマンドを使用する 341
 - penable コマンドを使用する 350
 - pshare コマンドを使用する 545
 - pstart コマンドを使用する 552
 - 使用不可
 - pdisable コマンドを使用する 342
 - phold コマンドを使用する 355
 - 遅延
 - 使用可能化 341
 - リスト 341
 - リスト
 - pdelay コマンドを使用する 341
 - penable コマンドを使用する 350
 - phold コマンドを使用する 355
 - pstart コマンドを使用する 552
 - 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 682
 - からのミラーの除去
 - rmlvcopy コマンドを使用する 816
 - ボリューム・グループからの除去
 - rmlv コマンドを使用する 814

A

- acct/* コマンド
 - 削除 700
 - nulladm 276
 - prctmp 449
 - prdaily 450
 - prtacct 511

B

- Bourne シェル shell 903

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 697

D

- Diablo 630 印刷ファイル
 - PostScript への変換
 - ps630 コマンドを使用する 541
- diff リスト 318

E

enotifyevent コマンド 229
enotifyevent スクリプト 229

F

fastboot コマンド 675

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 312

I

IMAP コマンド

pop3d 425
pop3ds 426

IPv6 NDP (隣接ディスカバリー・プロトコル) 14

K

Korn シェル

拡張 754
起動 754
制限付きの 754

M

MH

post コマンド 431
prompter コマンド 503
rcvpack コマンド 653
rcvstore コマンド 654
rcvtty 655

MultiPath I/O

rmpath コマンド 824

N

named デーモン 1
の説明 1

named8 デーモン 4
named9 デーモン 8
named-checkconf 2
named-checkzone 2
named-compilezone 2
namerslv コマンド 9
ncheck コマンド 12
nddctl コマンド 13

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 17

ndp コマンド 14

ndp デーモン 14

NDP (隣接ディスカバリー・プロトコル) 14

ndpd-host 15

ndpd-router デーモン 17

ndx コマンド 22

neqn コマンド 24

netcd デーモン 25

netcdctrl コマンド 27

netpmon コマンド 29

netrule コマンド 39

netstat コマンド 44

インターフェース表示 44

経路指定テーブル表示 44

Network Time Protocol コマンド

ntpdate 248

ntptrace 272

newaliases コマンド

メール 55

newform コマンド 56

newgrp コマンド 58

newkey コマンド

NIS 60

next コマンド 63

NFS コマンド

nfsstat 82

on 290

rmnfs 820

rmnfsexp 821

rmnfmnt 822

rpcgen 891

rpcinfo 892

rup 929

rusers 933

rwall 937

NFS デーモン

nfsd 69

pcnfsd 889

portmap 428

rexd 749

rstatd 912

rusersd 935

rwalld 938

nfs4cl コマンド 65

nfs4smctl 67

nfsauthreset 68

nfsd デーモン 69

nfshostkey 71

nfshostmap 72

nfso コマンド 72

nfsrgyd 81

nfsstat コマンド 82

nfs.clean コマンド 64

nice コマンド 86
NIM オブジェクト
操作を実行する
nim コマンドを使用する 88
NIM コマンド
nim 88
nimadapters 125
nimclient 140
nimconfig 144
niminit 151
nim_clients_setup 104
nim_master_recover 106
nim_master_setup 109
nim_update_all 123
nim コマンド 88
nimadapters 125
nimadm コマンド 131
nimclient コマンド 140
nimconfig コマンド 144
nimdef コマンド 147
niminit コマンド 151
niminv コマンド 155
nimol_backup コマンド 161
nimol_config コマンド 163
nimol_install コマンド 165
nimol_lslpp コマンド 168
nimol_update コマンド 169
nimquery 171
nim_clients_setup 104
nim_master_recover 106
nim_master_setup 109
nim_move_up コマンド 112
nim_update_all 123
NIS コマンド
newkey 60
rmkeyserv 811
rmyp 870
nistoldif 172
nlsrc コマンド 177
nm コマンド
オブジェクト・ファイル
記号テーブルの表示 179
nmon コマンド 182
no コマンド 203
nohup コマンド 227
notifyevent コマンド 229
notifyevent スクリプト 229
nroff コマンド 230
の数式テキストのフォーマット
neqn コマンドを使用する 24
nslookup コマンド 233
nsupdate コマンド 236
nsupdate4 コマンド 237
nsupdate8 コマンド 239
nsupdate9 241
ntpd4 デーモン 245

ntpddate コマンド 248
ntpddate4 250, 547
ntpdcc 253
ntpdcc コマンド 253
ntpqq コマンド 263
ntpqq4 デーモン 268
ntptrace コマンド 272
ntp-keygen4 259, 274
ntp-keygen4 コマンド 259, 274
nulladm コマンド 276

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 289
オブジェクト・クラスからの除去 287
オブジェクト・クラス内の変更 285
オブジェクト・クラスへの追加 283
オブジェクト・クラス
オブジェクトの除去 287
記述の表示 289
基準にマッチするオブジェクトの検索 289
コンパイル 285
削除 288
内のオブジェクトの変更 285
へのオブジェクトの追加 283
データベースを圧縮する 379
on コマンド 290
openpts 291
openpts コマンド 291
oslevel コマンド 298
ospf_monitor コマンド 300
OS_install コマンド 292

P

pac コマンド 303
pack コマンド 304
packf コマンド 306
pagdel 308
pagesize コマンド 309
paginit 310
paglist 311
panel20 コマンド 312
passwd コマンド 312
paste コマンド 316
patch コマンド 318
pathchk コマンド 323
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 324
pcat コマンド 340
pcnfsd デーモン 889

pdelay コマンド 341
 pdisable コマンド 342
 pdlink 343
 pdmkdir 345
 pdmode 346
 pdrmdir 348
 pdset 349
 penable コマンド 350
 perfwb コマンド 351
 phold コマンド 355
 pic コマンド 356
 ping コマンド
 の説明 365
 piobe コマンド 372
 pioburst コマンド 374
 piocvt コマンド 370, 376
 piodigest コマンド 377
 piofontin コマンド 380
 pioformat コマンド 381
 piofquote コマンド 383
 pioout コマンド 393
 piopredef コマンド 395
 pkgadd コマンド 396
 pkgask コマンド 399
 pkgchk コマンド 401
 pkginfo コマンド 403
 pkgmk コマンド 405
 pkgparam コマンド 407
 pkgproto コマンド 409
 pkgrm コマンド 411
 pkgtrans コマンド 413
 platform_dump 415
 plotgbe コマンド 417
 plotlbe コマンド 418
 pmctl コマンド 419
 pmcycles コマンド 421
 pmlist コマンド 422
 pmtu コマンド 424
 pop3d デーモン 425
 pop3ds デーモン 426
 portmap デーモン 428
 portmir コマンド 429
 post コマンド 431
 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 549
 印刷のページ範囲を選択する
 psrev コマンドを使用する 549
 Diablo 630 ファイルからの変換
 ps630 コマンドを使用する 541
 Tektronix 4014 ファイルからの変換
 ps4014 コマンドを使用する 540
 troff からの変換
 psroff コマンドを使用する 549
 troff 中間ファイルからの変換
 psc コマンドを使用する 542
 PostScript ファイル (続き)
 troff 中間ファイルからの変換 (続き)
 psdit コマンドを使用する 542
 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 383
 PPP 435
 PPP サブシステム
 始動と管理
 pppcontrold デーモン 435
 非同期接続 440
 RAS 情報を表示する
 pppstat コマンド 442
 pppattachd デーモン 432
 pppcontrold デーモン 435
 pppdial コマンド 440
 pppstat コマンド 442
 pprof コマンド 444
 pr コマンド 446
 praliases コマンド 448
 prctmp コマンド 449
 prdaily コマンド 450
 preparevsd コマンド 451
 preprnode コマンド 452
 prev コマンド 454
 printf コマンド 456
 probevctrl コマンド 460
 probevue コマンド 464
 proccred コマンド 469
 procfiles コマンド 470
 procflags コマンド 472
 procldd コマンド 474
 procmap コマンド 475
 procrun コマンド 479
 procsig コマンド 480
 procstack コマンド 482
 procstop コマンド 483
 proctree コマンド 484
 procwait コマンド 488
 procwdx コマンド 489
 proff コマンド 492
 projctl コマンド 494
 prompter コマンド 503
 proto コマンド 505
 proxymngr コマンド 506
 prs コマンド 508
 prtacct コマンド 511
 prtconf 513
 prtglbconfig コマンド 517
 ps コマンド 518
 ps4014 コマンド 540
 ps630 コマンド 541
 psc コマンド 542
 psdit コマンド 542
 pshare コマンド 545
 psplot コマンド 546

psrev コマンド 549
psroff コマンド 549
pstart コマンド 552
pstat コマンド 553
ptpd デーモン 555
ptsc 561
ptsc コマンド 561
ptsevt 562
ptsevt コマンド 562
ptsevtd 563
ptsevtd コマンド 563
ptx コマンド 563
pvi コマンド 567
pwchange コマンド 570
pwck コマンド 573
pwdadm コマンド 575
pwdck コマンド 577
pwtokey コマンド 581
pxed コマンド 584

Q

qadm コマンド 587
qcan コマンド 588
qchk コマンド 590
qdaemon コマンド 592
qhld コマンド 593
qmov コマンド 595
qosadd コマンド 596
qoslist コマンド 598
qosmod コマンド 599
qosremove コマンド 600
qosstat コマンド 601
qpri コマンド 603
qpri コマンド 605
qstatus コマンド 613
quot コマンド 617
quota コマンド 619
quotacheck コマンド 621
quotaoff コマンド 623
quotaon コマンド 623

R

raddbm コマンド 625
radiusctl コマンド 629
RAM ディスク 832
RAM ディスク、仮想
除去
 rmramdisk コマンドを使用する 832
raso コマンド 630
ras_logger コマンド 636
rbacqry コマンド 637
rbactoldif コマンド 642
rc コマンド 644

rcp コマンド 648
rcvdist コマンド
 着信メッセージ
 追加の宛先へのコピーの送信 653
 MH
 rcvdist コマンド 653
rcvpack コマンド 653
rcvstore コマンド 654
rcvtty コマンド 655
rc.mobip6 644
rc.powerfail コマンド 645
rc.wpars 648
rdist コマンド 656
rdistd コマンド 669
rdump コマンド 670
read コマンド 672
readlvcopy コマンド 674
reboot コマンド 675
rebootwpar コマンド 676
recfgct コマンド 677
recreatevg コマンド 679
recsh コマンド 681
redefinevg コマンド 682
reducevg コマンド 683
refer コマンド 685
refile コマンド 687
refresh コマンド 690
refrsrc コマンド 691
refsensor コマンド 693
rembak コマンド 698
remove コマンド 700
removevsd コマンド 701
remsh コマンド 905
rendev コマンド 702
renice コマンド 703
renice コマンドを使用した実行中のプロセスの優先順位の変更
 703
reorgvg コマンド 705
repl コマンド 706
replacepv コマンド 710
repquota コマンド 712
reset コマンド 713
resetsrc コマンド 714
resize コマンド 719
resource_data_input 情報ファイル 720
restart-secldapclntd 724
restore コマンド 726
restorevgfiles コマンド 737
restvg コマンド 739
restwpar コマンド 742
restwparfiles コマンド 744
resumevsd コマンド 746
rev コマンド 747
revnetgroup コマンド 748
rexed デーモン 749
rexec コマンド 750

rexec コマンド用のサーバー機能、TCP/IP 751
rexeed デーモン 751
rgb コマンド 752
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 17
ripquery コマンド 753
rksh コマンド 754
rlogin コマンド 757
rlogind デーモン 759
rmail コマンド 764
rmaudrec コマンド 766
rmauth コマンド 764
rmC2admin コマンド 770
rmCCadmin コマンド 771
rmccli マニュアル・ページ 772
rmcctrl コマンド 777
rmcdomainstatus コマンド 781
rmcifscred コマンド 784
rmcifsmnt コマンド 785
rmclass コマンド 786
rmcluster コマンド 787
rmcomg コマンド 788
rmcondition コマンド 790
rmcondresp コマンド 792
rmcosi コマンド 796
rmdel コマンド 797
rmdev コマンド 798
rmdir コマンド 801
rmdom コマンド 802
rmf コマンド 803
rmfilt コマンド 805
rmfs コマンド 806
rmgroup コマンド 807
rmiscsi コマンド 809
rmitab コマンド 810
rmkeyserv コマンド 811
rmlpcmd コマンド 812
rmlv コマンド 814
rmlvcopy コマンド 816
rmm コマンド 817
rmnamsv コマンド 819
rmnfs コマンド 820
rmnfsexp コマンド 821
rmnfsmnt コマンド 822
rmnfsproxy コマンド 823
rmpath コマンド 824
rmprtsv コマンド 827
rmpps コマンド 828
rmqos コマンド 829
rmque コマンド 830
rmquedev コマンド 831
rmramdisk コマンド 832
rmresponse コマンド 833
rmrole コマンド 836
rmrpdomain コマンド 837

rmrpnod コマンド 840
rmrset 843
rmrsrc コマンド 844
rmsecattr コマンド 848
rmsensor コマンド 850
rmsock コマンド 853
rmss コマンド 854
rmtcpip コマンド 861
rmts コマンド 862
rmtun コマンド 863
rmuser コマンド 865
rmusil コマンド 864
rmvfs コマンド 867
rmvirprt コマンド 868
rmwpar コマンド 869
rmyp コマンド 870
rmdc 871
rmdc-confgen コマンド 872
roffbib コマンド 874
rolelist コマンド 875
roleqry 876
rolerpt コマンド 878
rollback コマンド 880
route コマンド 881
routed デーモン 886
rpcgen コマンド 891
rpcinfo コマンド 892
rrestore コマンド 899
Rsh コマンド 903
rsh コマンド 905
rshd デーモン 909
rstatd デーモン 912
rsyslogd デーモン 912
rtcd デーモン 915
rt_enable コマンド 915
runacct コマンド 917
runact コマンド 921
runcat コマンド 925
runlpcmd コマンド 925
rup コマンド 929
ruptime コマンド 930
ruser コマンド 931
rusers コマンド 933
rusersd デーモン 935
rvsdrestrict コマンド 935
rwall コマンド 937
rwalld デーモン 938
rwho コマンド 938
rwhod デーモン 940

S

SCCS

 デルタ・ファイルを除去する

 rmdel コマンドを使用する 797

SCCS コマンド

prs 508
rmdel 797

SMIT

による印刷キューの作成 387
によるプリンターの作成 387
プリンター・ダイアログの構築 387

SRC

サブサーバー・オブジェクト定義の除去 853
サブシステム通知メソッドの除去 824
サブシステム・オブジェクト定義の除去 859

SRC 構成コマンド

rmnotify 824
rmserver 853
rmssys 859

T

TCP/IP

印刷サービス
構成解除 827
インターネット・ドメイン・ネーム・サーバーの照会 233
経路指定テーブル
手動入力の実行 881
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 931
サーバー機能 940
提供 759
デーモン
named 1
rexecd 751
パラメーター
チューニング 203
ホスト
ログインしているユーザーのリスト表示 938

TCP/IP smit コマンド

namerslv 9
rmnamsv 819
rmprtsv 827
ruser 931

TCP/IP コマンド

リモート・ホスト上での実行 750
namerslv 9
netstat 44
no 203
nslookup 233
rmnamsv 819
rmprtsv 827
route 881
ruser 931
rwho 938

TCP/IP デーモン

rlogind 759
routed 886
rshd 909
rwhod 940

Tektronix 4014 ファイル

PostScript への変換
ps4014 コマンドを使用する 540

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 719

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 542
psdit コマンドを使用する 542

troff ファイル

PostScript への変換
psroff コマンドを使用する 549

[特殊文字]

/etc/filesystems ファイル

rmfs コマンドを使用してエントリーを除去する 806

/etc/inittab ファイル

レコードの除去
rmitab コマンドを使用する 810

/etc/vfs ファイル

エントリーの除去
rmvfs コマンドを使用する 867



Printed in Japan

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