

AIX バージョン 7.2

**コマンド・リファレンス
第 6 巻 (v から z)**

IBM

AIX バージョン 7.2

**コマンド・リファレンス
第 6 巻 (v から z)**

IBM

お願い

本書および本書で紹介する製品をご使用になる前に、303 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 6, v - z

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	v
強調表示	v
AIX でのケース・センシティブ	v
ISO 9000	v
Single UNIX Specification のサポート	v
V	1
vacation コマンド	1
val コマンド (SCCS)	3
varyoffvg コマンド	4
varyonvg コマンド	6
vc コマンド	9
vgrind コマンド	12
vi または vedit コマンド	14
view コマンド	33
vmh コマンド	34
vmo コマンド	35
vmstat コマンド	50
vpdadd コマンド	64
vpddel コマンド	66
vsdata1st コマンド	67
vsdchgserver コマンド	69
vsdelnode コマンド	70
vsdelvg コマンド	72
vsdnode コマンド	73
vsdsk1st コマンド	75
vsdvg コマンド	78
vsdvgts コマンド	79
W	83
w コマンド	83
wait コマンド	84
wall コマンド	85
wallevnt コマンド	86
watch コマンド	88
wc コマンド	90
what コマンド	92
whatis コマンド	93
whatnow コマンド	94
whereis コマンド	98
which コマンド	100
which_fileset コマンド	100
who コマンド	102
whoami コマンド	105
whodo コマンド	105
whois コマンド	107
whom コマンド	109
wlmassign コマンド	111
wlmcheck コマンド	113
wlmcntrl コマンド	115
wlmstat コマンド	118

wol コマンド	123
wparerr コマンド	125
wparexec コマンド	126
wpar_reg_script コマンド	133
wparprnterr コマンド	134
write コマンド	134
writesrv デーモン	139
wtmpfix コマンド	140
wump コマンド	141
X	143
X コマンド	143
x_add_fs_fpe コマンド	155
x_add_nfs_fpe コマンド	156
x_rm_fpe コマンド	157
xargs コマンド	158
xauth コマンド	162
xclock コマンド	165
xcmsdb コマンド	167
xdm コマンド	168
xfindproxy コマンド	183
xfs コマンド	184
xget コマンド	187
xhost コマンド	189
xinit コマンド	190
xkbcomp コマンド	192
xkbevd デーモン	194
xkbprint コマンド	195
xlock コマンド	197
xlsfonts コマンド	199
xmbind コマンド	200
xmkmf コマンド	201
xmwlm コマンド	202
xmodem コマンド	203
xmodmap コマンド	206
xmpeek コマンド	208
xmscheck コマンド	210
xmtopas コマンド	211
xntpd デーモン	213
xntpd コマンド	216
xpr コマンド	224
xpreview コマンド	226
xprofiler コマンド	229
xrdb コマンド	231
xsend コマンド	234
xset コマンド	235
xsetroot コマンド	239
xss コマンド	240
xstr コマンド	241
xterm コマンド	243
xwd コマンド	269

xwud コマンド	270
y	273
yacc コマンド	273
yes コマンド	275
ypbind デーモン	276
ypcat コマンド	277
ypinit コマンド	279
ypmatch コマンド	280
yppasswd コマンド	281
yppasswdd デーモン	282
yppoll コマンド	284
yppush コマンド	285
ypserv デーモン	286
ypset コマンド	287

ypupdated デーモン	289
ypwhich コマンド	290
ypxfr コマンド	291

Z 295

zcat コマンド	295
zdump コマンド	296
zic コマンド	298

特記事項 303

プライバシー・ポリシーに関する考慮事項	305
商標	305

索引 307

本書について

本書は、AIX® オペレーティング・システムのコマンドについての詳細情報を網羅して、エンド・ユーザーの皆様にお届けするものです。コマンドはアルファベット順およびカテゴリー別にリストされており、コマンドおよびそれらの使用可能なフラグについての完全な記述が示されています。利用可能な場合、各コマンド・リストには例が含まれています。本書 (第 6 巻) には、先頭文字が **v** から **z** までの AIX コマンドが記載してあります。なお、本資料は、オペレーティング・システムに付属して配送されている文書 CD にも収められています。

強調表示

本書では、次の強調表示規則を使用しています。

太字	コマンド、サブルーチン、キーワード、ファイル、構造体、ディレクトリー、およびシステムによって名前が事前に定義されているその他の項目を表します。さらに太字の強調表示は、ユーザーが選択するボタン、ラベル、およびアイコンなどのグラフィカル・オブジェクトも示します。
イタリック	ユーザーが入力する実際の名前または値のパラメーターを示します。
モノスペース	具体的なデータ値の例、表示される可能性があるテキストの例、プログラマーとして作成する可能性があるものに似たプログラム・コードの一部の例、システムからのメッセージ、またはユーザーが入力しなければならないテキストを示します。

AIX でのケース・センシティブ

AIX オペレーティング・システムでは、すべてケース・センシティブとなっています。これは、英大文字と小文字を区別するという意味です。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「is not found」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。予期しない処理が実行されないように、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。

V

以下の AIX コマンドは、文字 *v* で始まります。

vacation コマンド

目的

メールの受信側が不在であることを示すメッセージを送信側に戻します。

構文

```
vacation [ { -I | User } ] | [ { -f Number [ Unit ] | User } ]
```

説明

vacation コマンドは、メール・メッセージの送信側にメッセージを戻して、受信側が不在であることを送信側に通知します。この使用目的は **\$HOME/.forward** ファイルの中に入っており、メッセージを受信しながら一方でメッセージを送信側に戻すことができます。

注: Sendmail バージョン 8.9.3 およびそれ以降のリリースでは、以下のいずれかの条件が存在する場合、**.forward** ファイルを無視するというセキュリティ上の拡張が行われています。

- **.forward** ファイルは、グループまたはワールド書き込み可能許可を持ちます。
- **.forward** ファイルのどの親ディレクトリーも、グループまたはワールド書き込み可能許可を持ちます。

.forward ファイルが無視されたため、**vacation** プログラムが作動していないと考えられる場合は、許可を確認してください。**.forward** ファイルのどの親ディレクトリーにもグループまたはワールド書き込み可能許可を持たせる必要がある場合は、**sendmail** 構成ファイル内の **DontBlameSendmail** オプションに正しい値を設定してください。

vacation コマンドの実行にあたっては、各送信側に戻すメッセージが **\$HOME/.vacation.msg** ファイルに入っていることを前提とします。このファイルが存在しない場合、**vacation** コマンドは、システム全体のデフォルトの不在メッセージ・ファイルである **/usr/share/lib/vacation.def** を探します。この場合、発信側 (From) または件名 (Subject) などの必要なヘッダーをすべて含む完全なメッセージとなります。デフォルトでは、このメッセージは、メールを送信するユーザーに 1 週間に一度だけ送信されます。メッセージを送信する間隔を変更するには、**-f** フラグを使用します。メッセージを送信するユーザーの名前は、**\$HOME/.vacation.pag** ファイルおよび **\$HOME/.vacation.dir** ファイルに保存されます。これらのファイルは、**-I** (i の大文字) フラグで、**vacation** コマンドが受信側のユーザー ID 用に初期化されるときに作成されます。

-I フラグを指定しないと、**vacation** コマンドは、From 行の標準入力から最初の行を読み取り、送信側を特定します。標準入力に使用できるテキストがなければ、このコマンドはエラー・メッセージを戻します。正しくフォーマットされた着信メールにはすべて From 行があります。From ヘッダー行からメッセージが Postmaster である MAILER-DAEMON からのものであると判明したり、最初の From 行に REQUEST@ という文字列があったり、ヘッダー Precedence: bulk または Precedence: junk 行があると、メッセージは送信されません。

フラグ

項目	説明
-I	\$HOME/vacation.pag ファイルおよび \$HOME/vacation.dir ファイルを初期化します。 \$HOME/forward ファイルを変更する前に、このフラグを使用して vacation コマンドを実行します。
-fNumber [Unit]	不在メッセージを繰り返し送信する間隔を指定します。 <i>Number</i> パラメーターは整数値で、 <i>Unit</i> パラメーターは時間の単位を指定します。 <i>Unit</i> パラメーターには、以下のいずれかの値を指定できます。
s	秒数
m	分数
h	時間数
d	日数
w	週数

注: **-f** フラグを **-I** フラグとともに使用することはできません。

例

- 受信側が不在であることを告げるメッセージを送信側に戻すには、**vacation** コマンドを使用する前に、**\$HOME/vacation.pag** ファイルと **\$HOME/vacation.dir** ファイルを初期化する必要があります。これらのファイルを初期化するには、次のように入力します。

```
vacation -I
```

- .forward** ファイルを変更します。例えば、Mark は、次のステートメントを **.forward** ファイルに入力します。

```
mark,|"/usr/bin/vacation mark"
```

送信側は、**\$HOME/vacation.msg** ファイルに入っているメッセージを受信するか、それがなければ、**/usr/share/lib/vacation.def** ファイルにあるデフォルト・メッセージを受信します。どちらのファイルも存在しなければ、メール・メッセージの送信側には自動的にメッセージは送信されず、エラー・メッセージは生成されません。この 2 つのファイルのどちらかが存在する場合、送信側から mark に送信されるメッセージ数に関係なく、送信側は 1 週間に一度 mark からの不在メッセージを 1 件受信します。

- .forward** ファイルに次のエントリーが入っているものと仮定します。

```
mark,|"/usr/bin/vacation -f10d mark"
```

送信側は、mark に送信するメッセージ数に関係なく、10 日ごとに mark から不在メッセージを 1 件受信します。

- デフォルトの不在メッセージと違うメッセージを作成するには、**\$HOME/vacation.msg** ファイルを作成し、このファイルにメッセージを追加します。次の例は、不在メッセージを示しています。

```
From: mark@odin.valhalla (Mark Smith)
Subject: I am on vacation.
Delivered-By-The-Graces-Of: the Vacation program
I am on vacation until October 1. If you have something urgent,
please contact Jim Terry <terry@zeus.valhalla>.
--mark
```

- 不在メッセージを取り消すには、**\$HOME** (ログイン) ディレクトリーから **.forward** ファイル、**.vacation.dir** ファイル、**.vacation.pag** ファイル、および **.vacation.msg** ファイルを除去します。

```
rm .forward .vacation.dir .vacation.pag .vacation.msg
```

ファイル

項目	説明
<code>\$HOME/.forward</code>	受信側がメールを転送したい相手の名前が入っています。
<code>/usr/share/lib/vacation.def</code>	システム全体のデフォルトの vacation メッセージが入っています。
<code>\$HOME/.vacation.dir</code>	vacation コマンドの使用中に、受信側にメールを送信してきたユーザーの名前が入っています。
<code>\$HOME/.vacation.msg</code>	受信側の個人の vacation メッセージが入っています。
<code>\$HOME/.vacation.pag</code>	vacation コマンドの使用中に、受信側にメールを送信してきたユーザーの名前が入っています。
<code>/usr/bin/vacation</code>	vacation コマンドが入っています。

関連情報:

.forward ファイル

不在メッセージ通知の送信

val コマンド (SCCS)

目的

SCCS ファイルの妥当性を検査します。

構文

```
val [ -s ] [ -rSID ] [ -mName ] [ -yType ] File ...
```

説明

val コマンドは、指定されたファイルを読み取り、そのファイルが付随しているフラグで指定された特性に合うソース・コード制御システム (SCCS) ファイルであるかどうかを判別します。 *File* 値に - (マイナス) を指定した場合、**val** プログラムは、標準入力を読み取り、標準入力の各行を **val** フラグおよび SCCS ファイル名として解釈します。ファイル終わり文字によって入力が終了します。

val コマンドは、処理する各ファイルに対して、メッセージを標準出力に表示します。

フラグ

各名前付きファイルには、各フラグまたはフラグ・グループが個別に適用されます。フラグの指定順序は任意です。

項目	説明
-m <i>Name</i>	<i>Name</i> 値を、指定されたファイル内の SCCS 31 識別キーワードと比較します。
-r <i>SID</i>	妥当性検査をするファイルの <i>SID</i> を指定します。 <i>SID</i> は有効でなければならず、あいまいであってはなりません。
-s	通常は標準出力に書き出されるエラー・メッセージを抑制します。
-yType	指定されたファイル内の SCCS 識別キーワードと比較する型を指定します。

終了状況

val コマンドは、すべてのファイルに対して正常に終了した場合は 0 を戻します。それ以外の場合は、起こりうるエラーを示す 8 ビット・コードを戻します。この 8 ビット・コードは、ビット列として解釈されます。そこで設定されるビット (左から右に) は以下の意味を表します。

項目	説明
0x80	脱落しているファイル引数
0x40	未知または重複しているオプション
0x20	破壊された SCCS ファイル
0x10	ファイルをオープンできないか、または SCCS 以外のファイル
0x08	SID が無効またはあいまい
0x04	SID が存在しない。
0x02	, y がミスマッチ
0x01	3l, m がミスマッチ

注: **val** コマンドは、規定のコマンド・ラインで複数のファイルを処理し、複数のコマンド・ラインも処理できます (標準入力から読み取るとき)。このような場合、戻されるコードは、各コードを総計したものです。つまり、各コマンド・ラインとファイルに対して生成されたコードに論理 OR が処理されます。

例

s.test.c ファイルが SCCS テキスト・ファイルであるかどうかを判別するには、次のように入力します。

```
val -ytext s.test.c
```

関連情報:

その他の SCCS コマンドのリストのセクション

get コマンド

varyoffvg コマンド

目的

ボリューム・グループを活動停止します。

構文

```
varyoffvg [ -s ] VolumeGroup
```

説明

varyoffvg コマンドは、*VolumeGroup* パラメーターで指定したボリューム・グループとその関連する論理ボリュームを活動停止します。最初に論理ボリュームをクローズする必要があります。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、アンマウントする必要があります。

ボリューム・グループを活動化するには、**varyonvg** コマンドを使用します。

注: このコマンドを使用するには、**root** ユーザー権限を持っているか、またはシステム・グループのメンバーである必要があります。

System Management Interface Tool (SMIT) を使用して、このコマンドを実行できます。SMIT を使用するには、次のように入力します。

```
smit varyoffvg
```

注:

- ページング・スペース・ボリュームのあるボリューム・グループは、ページング・スペースが活動状態にある間はオフに変えることはできません。活動状態のページング・スペース・ボリュームのあるボリ

ユーモ・グループを活動停止するときには、事前にシステムの初期化時にページング・スペースが自動的に活動化されていないことを確認してから、そのシステムをリブートします。

- `varyoffvg` コマンドは、ボリューム・グループに対して実行中のバックグラウンド・スペース再利用プロセスをすべて破棄します。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。

フラグ

項目	説明
<code>-s</code>	ボリューム・グループ内で論理ボリューム・コマンドだけが使用できるように、ボリューム・グループをシステム管理モードにします。このモードでは、ユーザーが論理ボリュームをオープンしたり、アクセスすることはできません。

終了状況

このコマンドは、下記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. ボリューム・グループ `vg03` を活動停止するには、次のように入力します。

```
varyoffvg vg03
```

2. ボリューム・グループ `vg02` は活動停止するが、論理ボリューム・コマンドの効力を残すには、次のように入力します。

```
varyoffvg -s vg02
```

ボリューム・グループ内の論理ボリュームをオープンすることはできませんが、論理ボリューム・コマンドの効力は継続します。

ファイル

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

関連資料:

6 ページの『`varyonvg` コマンド』

関連情報:

ページング・スペースの概念

RBAC

Trusted AIX

varyonvg コマンド

目的

ボリューム・グループを活動開始します。

構文

```
varyonvg [ -b ] [ -c ] [ -f ] [ -M ltgsize ] [ -n ] [ -p ] [ -r ] [ -s ] [ -t ] [ -u ] [ -k loc|rem ] [ -d ] [ -o ] [ -O ] volumegroup
```

説明

varyonvg コマンドは、*volumegroup* パラメーターで指定したボリューム・グループとそのすべての関連する論理ボリュームを活動化します。活動化されたボリューム・グループは使用可能になります。ボリューム・グループが活動化されると、現在の物理区画以外の物理区画は同期化されます。PVMISSING 状態にあり置き換えられた物理ボリュームを、**varyonvg** コマンドを使用して PACTIVE 状態に戻すことができます。

注: 物理ボリュームがダンプ・デバイスの一部である場合、**varyonvg** コマンドはそのボリュームを PACTIVE 状態に戻すことができません。このコマンドを適切に実行させるには、一時的にダンプ・デバイスを変更してください。

デバイス構成データベースと論理ボリューム・マネージャー (LVM) に保管された情報に矛盾があると、全物理ボリュームのリストがその状況とともに標準出力に表示されます。ボリューム・グループはオンに構成変更されていても、そうでなくてもかまいません。リストを注意深く検査し、表示された状況の内容によっては、システム保全のために適切な処置をとる必要があります。

並行モードでオンに構成変更する間に、システムに未知の論理ボリュームがあることを varyon プロセスが検出した場合、その定義がインポートされます。新規デバイス・スペシャル・ファイルの権限および所有権が、ボリューム・グループ・スペシャル・ファイルの権限および所有権に複製されます。論理ボリュームが作成されたノード上で論理ボリュームのデバイス・スペシャル・ファイルの権限および/または所有権を変更すると、このノードで同じ変更を実行する必要があります。

制約事項: 従来の並行モードは AIX 5.3 ではサポートされていません。

物理ボリュームのほとんどが欠損しているためにボリューム・グループ をオンに構成変更できない場合、すべての物理ボリューム・リストがその状況とともに表示されます。この状態でボリューム・グループ をオンに構成変更するには、強制オプションを使う必要があります。

大多数の物理ボリュームがアクセス可能でない (規定数がない) 場合には、**varyonvg** コマンドはボリューム・グループをオンに構成変更することはできません。このことは、規定数検査が使用不可になっている場合であっても真です。規定数検査を使用不可にすることは、規定数が欠損している場合であっても、ボリューム・グループがオンに構成変更されたままになっていることを保証することになります。

PV_MISSING 状態の物理ボリュームが存在し、規定数検査が無効になっている場合、ボリューム・グループはオンに構成変更されません。このことは、規定数のディスクが使用可能であっても真です。この状態でオンに構成変更するためには、強制オプションを使用するか、または環境変数 MISSINGPV_VARYON を TRUE に設定します (ブート時に存在していないディスクでボリューム・グループを変更する必要がある場合には、この値を `/etc/environment` で設定します)。

上記の場合 (強制でオンに構成変更するオプションを使用するか MISSINGPV_VARYON 変数を使用する場合) には、 ボリューム・グループの保全性についての全責任はユーザーにあります。

要件: このコマンドを使用するには、 root ユーザー権限を持っているか、または **system** グループのメンバーでなければなりません。

System Management Interface Tool (SMIT) を使用して、このコマンドを実行できます。 SMIT を使用するには、次のように入力します。

```
smit varyonvg
```

フラグ

項目	説明
-b	<p>標準 varyonvg コマンドの結果としてロックされたディスク上のディスク予約を切断します。既にオンに構成変更されているボリューム・グループ上のこのフラグを使用します。</p> <p>注:</p> <ul style="list-style-type: none">このフラグは、既定のボリューム・グループ内のすべてのディスクをアンロックします。-b フラグは、SC_FORCED_OPEN フラグを使用してボリューム・グループ内のディスクをオープンします。SCSI および FC ディスクの場合、このフラグは、そのディスクが存在するターゲット・アドレス上のすべての LUNS を強制的にオープンします。したがって、varyon -b オプション使用時は、ボリューム・グループ間でターゲット・アドレスを共有すべきではありません。アクティブ・ページング・スペースを含むボリューム・グループ上で -b フラグを使用すると、システムをハングさせることがあります。
-c	<p>ボリューム・グループを Enhanced Concurrent (拡張並行) モードでオンに構成変更します。これは、ボリューム・グループが Concurrent Capable (並行機能対応) または Enhanced Concurrent Capable (拡張並行機能対応) であり、システムに PowerHA® SystemMirror® 製品がロードされていて使用可能である場合にのみ、行うことができます。どちらにも当てはまらない場合は、ボリューム・グループは varyon に失敗します。</p> <p>要件: 拡張並行ボリューム・グループは、グループ・サービスを使用します。Group Services (グループ・サービス) は、このモードでボリューム・グループを活動化する前に構成しておく必要があります。</p>
-d	<p>データの相違を認めます。ユーザーがボリューム・グループをオンラインにしようとする際に、その対向サイトのキャッシュには非ミラーリング更新データが含まれている可能性があり、またそのキャッシュにアクセスできない場合のみ、-d フラグは有効です。ユーザーがバックレベル・データを使用する可能性があり、また -d フラグを指定していないことを varyonvg コマンドが検出すると、このコマンドは重大なエラー・メッセージを出して失敗します。</p>
-f	<p>Geographic LVM の非同期ミラーリングについての詳細は、「<i>Geographic Logical Volume Manager for PowerHA SystemMirror Enterprise Edition</i>」を参照してください。</p> <p>現在、規定数の使用可能なディスクを持っていないボリューム・グループをアクティブにすることを可能にします。アクティブ状態にすることのできないディスクはすべて、除去された状態にされます。ボリューム・グループで、最低 1 つのディスクが使用可能である必要があります。 -f フラグ (クォーラム脱落をオーバーライドするために使用) は、ボリューム・グループがクォーラムをなくしていなければ無視されます。ディスクが除去済み状態にある場合、そのディスクを再びアクティブ状態にするには chpv -v a PVname コマンドを使用します。</p>
-k loc rem	<p>ローカル・ミラー・コピーまたはリモート・ミラー・コピーのデータを保持します。以下の属性を -k フラグと一緒に指定することができます。</p> <p>loc ローカル・ミラー・コピー・データを保存します。ローカルとは、1 次サイトではなく、ローカル物理ボリュームのことです。</p> <p>rem リモート・ミラー・コピー・データを保存します。リモートとは、リモート・サイトではなく、リモート物理ボリュームのことです。</p>

Geographic LVM の非同期ミラーリングについての詳細は、「*Geographic Logical Volume Manager for PowerHA SystemMirror Enterprise Edition*」を参照してください。

項目	説明
-M <i>ltgsize</i>	<p>ボリューム・グループの <i>ltgsize</i> を静的に設定します。 <i>ltgsize</i> の有効値には、128K、256K、512K、1M、2M、4M、8M、16M、32M、および 128M が含まれます。ボリューム・グループ内に <i>ltgsize</i> の最大転送容量以上で構成されていないディスクがある場合、 <i>varyonvg</i> コマンドは機能しません。</p>
-n	<p><i>volumegroup</i> パラメーター内の失効した物理区画の同期を使用不可にします。</p>
-o	<p>選択したコピー内では古くなっているが他のコピー内では最新である区画からのデータの使用を認めます。ローカル・コピーとリモート・コピーでデータに相違がある場合に、いずれかのコピーを保存するために -k フラグを指定すると、 varyonvg コマンドは失敗します。また、選択したローカル・コピーまたはリモート・コピー内の一部の区画は最新ではないので、 varyonvg コマンドが完全なコピーを保存することはできません。選択したコピー内では古くなっているが他のコピー内では最新である区画からのデータを使用するために、 -o フラグを指定することによって、コマンドの失敗を無効にすることができます。 -o フラグは、 -k フラグと一緒に指定する場合のみ有効です。</p>
-p	<p>Geographic LVM の非同期ミラーリングについての詳細は、「<i>Geographic Logical Volume Manager for PowerHA SystemMirror Enterprise Edition</i>」を参照してください。</p>
-r	<p>すべての物理ボリュームは、 varyonvg コマンドを使用するために、使用可能である必要があります。ボリューム・グループを読み取り専用モードでオンに構成変更します。このモードでは、下記のことはできません。</p> <ul style="list-style-type: none"> • 論理ボリュームへの書き込み操作 • LVM メタデータ更新 • 古い区画の同期
-s	<p>制約事項: 読み取り専用論理ボリュームへの JFS ファイルシステムのマウントは、サポートされません。</p> <p>制約事項: LVM メタデータ更新を必要とするすべての LVM 高水準コマンドは、このモードでの要求には失敗します。</p> <p>ボリューム・グループをシステム管理モードでのみ使用可能にします。論理ボリューム・コマンドはボリューム・グループ上で実行できますが、入出力用に論理ボリュームをオープンすることはできません。</p> <p>制約事項: また論理ボリューム・コマンドでは、 -s フラグを使用してオンに構成変更したボリューム・グループの論理ボリュームから読み取ったり、そのボリュームに書き込んだりすることはできません。 (chvg または mklvcopy のような) -s フラグを使用してオンに構成変更したボリューム・グループ内の論理ボリュームに書き込もうとすると、論理ボリュームは、エラー・メッセージを表示して、論理ボリュームへの書き込みまたはそのボリュームからの読み取りあるいはその両方ができなかったことを通知します。</p>
-t	<p>デバイス構成データベースおよび論理ボリューム・マネージャー内のタイム・スタンプを検査します。タイム・スタンプに矛盾がある場合は、データベースを同期化するために synclvodm コマンドが出されます。</p> <p>ヒント: 並行モードでボリューム・グループがオンに構成変更される場合は常に、この検査が行われません。</p>
-u	<p>ボリューム・グループをオンに構成変更しますが、ボリューム・グループを構成するディスクをアンロックされた状態のままにします。休止ボリューム・グループへの初回の <i>varyon</i> 操作の一部として、このフラグを使用します。</p>
-O	<p>ボリューム・グループが別のいずれかのノードでオンに構成変更されているかどうかにかかわらず、そのボリューム・グループに対し強制的に <i>varyon</i> 操作を行います。</p> <p>注: AIX 61 TL8 およびそれ以降のリリースにおいては、作成されたボリューム・グループが複数のノードにおいて非並行モードで <i>varyon</i> を受けられないようになっている場合、 varyonvg コマンドはボリューム・グループが <i>varyon</i> された状態で、LVM メタデータおよび ODM を更新します。 <i>varyon</i> 時に、 varyonvg コマンドはこのデータを読み取り、そのボリューム・グループが既に他のノードでオンに構成変更されている場合は機能停止する。 varyoffvg コマンドは、 <i>varyoff</i> 時に、ボリューム・グループの <i>varyon</i> 状態をリセットします。ボリューム・グループをオフに構成変更する前にシステムがクラッシュするか、ボリューム・グループが強制的にオフにされる場合、システムのリポート後に varyonvg コマンドは失敗します。このシナリオでは、ボリューム・グループへの <i>varyon</i> 操作を強制的に行うために -O フラグを使用します。</p>

重要: LVM の基本設計では、1 つの開始プログラムだけがボリューム・グループにアクセスできると仮定しています。PowerHA SystemMirror 製品は、共用ボリューム・グループのマルチノード・アクセスを同期化するために、LVM で作動します。しかし、多重開始プログラムのノードは、PowerHA SystemMirror を使用せずに、フラグが **-b** および **-u** のボリューム・グループに簡単にアクセスすることができます。ユーザーは、ボリューム・グループの状況情報が、この 2 つのフラグで迂回されているディスク保護 (ロック) の結果、妥協されるか、または不可解に変更されることがあるのを知っている必要があります。**-b** および **-u** フラグを使用する場合、データおよび状況出力の整合性を保つように保証することはできません。

終了状況

このコマンドは、下記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。

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. ボリューム・グループ **vg03** を活動化するには、次のように入力します。

```
varyonvg vg03
```

2. 現在の区画でない区画を同期化せずにボリューム・グループ **vg03** を活動化するには、次のように入力します。

```
varyonvg -n vg03
```

ファイル

項目	説明
/usr/sbin	varyonvg コマンド・ディレクトリーが入っています。
/tmp	コマンド実行時に一時ファイルを保管します。

関連資料:

4 ページの『**varyoffvg** コマンド』

関連情報:

chvg コマンド

RBAC

Trusted AIX

vc コマンド

目的

識別キーワードの代わりに割り当て値を使用します。

構文

vc [**-a**] [**-t**] [**-s**] [**-cCharacter**] [*Keyword=Value*]...

説明

vc コマンドは、標準入力から標準出力に行をコピーします。コマンド・ライン上のフラグとキーワード、入力内の制御ステートメントによって出力結果が変わります。**vc** コマンドは、ユーザーが宣言したキーワードをコマンド・ライン上に割り当てられた値と置き換えます。キーワードはテキスト内でも制御ステートメント内でも置き換えられます。

制御ステートメント

制御ステートメントは、制御文字で始まる 1 行のステートメントです (デフォルトの制御文字は **:** (コロン) です)。制御ステートメントによって入力の条件付き処理が指定されます。使用できる制御ステートメントのタイプは、以下のとおりです。

:if *Condition*

Text

項目	説明
:end	条件が真である場合に限り、 :if ステートメントと対応する :end との間の行をすべて標準出力に書き出します。 :if と :end のステートメントはネストできます。しかし、一度条件が偽になると、ネストされた残りの :if と :end のステートメントは、すべて無視されます。条件文の構文および使用できる演算子については、条件文の構文のセクションを参照してください。
:dcl <i>Keyword</i> , [<i>Keyword</i> . . .] :asg <i>Keyword=Value</i>	指定したキーワードを宣言します。すべてのキーワードを宣言しなければなりません。指定した値を指定したキーワードに割り当てます。 :asg ステートメントは、 vc コマンド・ラインのキーワード割り当てに優先します。後の :asg ステートメントは、関連したキーワードのそれ以前の割り当てをすべてオーバーライドします。宣言されても値 が割り当てられていないキーワードには、NULL 値が入ります。
:: <i>Text</i>	先行制御文字を 2 文字除去し、キーワードをそれぞれの値に置き換えて、その行を標準出力にコピーします。
:on または :off	すべての行においてキーワード置換をオンまたはオフにします。
:ctl <i>Character</i>	制御文字を <i>Character</i> 値に変更します。
:msg <i>Message</i>	標準エラー出力に、次のフォーマットでメッセージを書き込みます。 <i>Message(n)</i> : message where n is number of the input line on which the message appeared.
:err <i>Message</i>	エラー・メッセージを標準エラーに書き込みます。 vc コマンドは処理を停止し、終了値の 1 を戻します。エラー・メッセージのフォーマットは、次のとおりです。 ERROR: message ERROR: err statement on line n (vc15)

条件文の構文

使用できる項目とステートメントは、以下のとおりです。

condition	::=OR statement ::=NOR statement
OR statement	::=AND statement ::=AND statement OR statement
AND statement	::=expression ::=expression & AND statement
expression	::=(OR statement)

```

operator value ::=value operator value
               ::= = or != or < or >
               ::= ASCII string
               ::= numeric string

```

使用できる条件演算子とその意味は、以下のとおりです。

項目	説明
=	等しい
!=	等しくない
&	AND
&	OR
>	よりも大きい
<	よりも小さい
()	論理グループ化に使用
NOT	if の直後にだけ存在し、これがあると完全条件の値が反転されます。

> および < (大なり記号および小なり記号) は、符号なし整数値にのみ使用できます。例えば、012 > 12 は偽です。他のすべての演算子は文字列を修飾子として処理します。例えば、012 = 12 は真です。演算子の優先順位は、以下のとおりです。

- = != > < (すべて同じ優先順位)
- &
- &|

括弧を使用して優先順位を変更できます。

値と演算子や括弧の間には、ブランクかタブを少なくとも 1 つ入れなければなりません。

キーワード置換

キーワードの最初と最後は、制御ステートメントに使用した制御文字と同じものでなければなりません。キーワードは最高で 9 桁の英数字です。最初の文字は英字でなければなりません。キーワード値はどのような ASCII 文字列であってもかまいません。数字キーワード値は、符号なし数字文字列です。値にタブやスペースは使用できません。

フラグ

項目	説明
-a	テキスト行全体に渡って、制御文字で囲まれたキーワードを、その割り当てられた値と置き換えます (2 つの制御文字で始まるものに限定しない)。
-cCharacter	Character 値を制御文字として使用します。Character パラメーターには、ASCII 文字を指定しなければなりません。
-s	通常は標準エラーに表示される警告メッセージを表示しません。
-t	制御ステートメントを検出するために、行頭から最初のタブ文字まで (タブ文字を含む) のすべての文字を無視します。vc コマンドは、制御文字を検出すると、タブまで (タブを含む) のすべての文字を無視します。

終了状況

このコマンドは、下記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. *Keyword=Value* 割り当ての例を次に示します。

```
numlines=4
prog=acctg
pass4=yes
```

vc コマンドは、標準出力にテキストを書き込むときに、2 つの制御文字でマークされた入力テキスト行からすべての制御文字とキーワードを除去します。

2. 制御文字を解釈しないようにするには、次のように制御文字の前に円記号を置きます。

```
::the :prog: program includes several of the following¥:
```

:prog: キーワードはその値によって置き換えられますが、**¥:** は **:** (コロン) として標準出力に渡されます。

円記号 (¥) で始まり制御文字が後に続く入力行は、制御行として処理されず、円記号を取り除いて標準出力にコピーされます。これに対して、**vc** コマンドは、円記号で始まり、後に制御文字がない行は何も変更せずに書き込みます (先頭の円記号もそのまま)。

ファイル

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

関連情報:

その他の SCCS コマンドのリストのセクション

vgrind コマンド

目的

プログラム・リストを読みやすいようにフォーマットします。

構文

```
vgrind [ -f ] [ -n ] [ -t ] [ -x ] [ -PPrintdev ] [ -TName ] [ - ] [ -dFile ] [ -h Header ] [ -lLanguage ]
[ -sSize ] [ File ... ]
```

説明

vgrind コマンドは、*File* パラメーターで指定したプログラム・ソースを、**troff** コマンドを使用して読みやすいスタイルにフォーマット化します。コメントはイタリック体、キーワードは太字で、また現在の関数名は各ページごとそのマージンにリストされます。

vgrind コマンドは、フィルター・モードでも標準モードでも実行できます。

フィルター・モードの場合には、**vgrind** コマンドは、**tbl** コマンドによく似た形式のフィルターとして動作します。標準入力、以下の **troff** 式のマクロによって括弧に囲まれた行を除き、標準出力に直接渡されます。

項目	説明
.vS	処理を開始します。
.vE	処理を終了します。

先行する行は、**vgrind** コマンドの規則に従ってフォーマットされます。このフィルターからの出力は、出力用として **troff** コマンドに渡されます。**eqn** コマンドや **tbl** コマンドの順序に決まりはありません。

標準モードの場合には、**vgrind** コマンドは入力ファイルを受け取り、それらを処理し、正しい順序で **troff** コマンド、適切な後処理プログラム、およびプリンターに渡します。

どちらのモードでも、**vgrind** コマンドは小数点で始まる行は変更せずに渡します。

vgrind コマンドは、標準言語定義ファイル `/usr/share/lib/vgrindefs` または **-d** フラグで指定した代替ファイルのいずれかに定義した ASCII キーワードしかサポートしません。

フラグ

項目	説明
-f	強制的にフィルター・モードにします。
-n	強制的にキーワードのボールド化を停止します。
-t	フォーマットされたテキストが標準出力に渡されます。
-x	索引ファイルを読みやすいフォーマットで出力します。索引ファイル自体は、現行ディレクトリー内の index ファイルを指定して vgrind コマンドを実行すると作成されます。関数定義の索引は、 -x フラグと <i>File</i> パラメーターを指定して vgrind コマンドを実行すると作成されます。
-PPrintDev	qprt コマンドを使用して、出力データを <i>Printdev</i> に送信します。このフラグを指定しない場合は、 PRINTER 環境変数が使用されます。 PRINTER 環境変数が設定されていない場合は、システム・デフォルトが使用されます。
-TName	<i>Name</i> パラメーターで指定した troff デバイス用の出力データを作成します。この出力は、適切な後処理プログラムによって送信されます。デフォルトは ibm3816 後処理プログラムです。
-	入力を標準入力から強制的に取り込みます (-f フラグを指定するとデフォルトとなります)。
-dFile	代替の言語定義ファイルを指定します (デフォルトは <code>/usr/share/lib/vgrindefs</code> ファイル)。
-h Header	すべての出力ページに、特定のヘッダーを指定します (デフォルトはファイル名です)。 注: -h フラグと <i>Header</i> 変数の間にはブランク・スペースが 1 つ必要です。
-lLanguage	使用する言語を指定します。現在使用できる言語は、以下のとおりです。
c	C (デフォルト)。行上の関数名の前にはスペース、タブ、またはアスタリスクしか入れることができません。括弧付きのオプションも同じ行になければなりません。
csH	CSH。
p	PASCAL。関数名は、 function キーワードまたは procedure キーワードと同じ行になければなりません。
m	MODEL。関数名は、 isbeginproc キーワード句と同じ行になければなりません。
sh	SHELL。
r	RATFOR。
mod2	MODULA2。
yacc	YACC。
isp	ISP。
I	ICON。
-s Size	出力に使用するポイント・サイズを指定します (.ps 要求とまったく同じ)。

ファイル

項目 索引	説明
<code>/usr/bin/vgrind</code>	索引用のソースが作成されるファイルが入っています。 vgrind コマンドが入っています。
<code>/usr/share/lib/tmac/tmac.vgrind</code>	マクロ・パッケージが入っています。
<code>/usr/share/lib/vfontedpr</code>	プリプロセッサが入っています。
<code>/usr/share/lib/vgrindefs</code>	言語記述が入っています。

関連情報:

qprt コマンド

tbl コマンド

troff コマンド

vgrindefs ファイル・フォーマット

vi または vedit コマンド

目的

ファイルをフルスクリーン表示で編集します。

構文

```
{ vi | vedit } [ -l ] [ -R ] [ -tTag ] [ -v ] [ -wNumber ] [ -yNumber ] [ -r [ File ] ] [ { + | -c } { Subcommand } ] [ File ... ]
```

説明

vi コマンドは、基本 **ex** エディターに基づいてフルスクリーン・エディターを始動します。したがって、**vi** エディター内では **ex** サブコマンドが使用できます。**vedit** コマンドは、**vi** エディターの初心者向けバージョンを始動します。**vedit** エディターでは、**report** オプションが 1 に設定され、**showmode** オプションが設定され、**novice** オプションが設定されて、ライン・エディターとなります。

編集するファイル名 (複数可) を指定して **vi** エディターを始動します。コマンド・ラインで複数の *File* パラメーターを指定すると、**vi** エディターは各ファイルを指定順に編集します。既存のファイル上の **vi** エディターは、ファイル名、行番号、および画面最下部の文字数を表示します。マルチバイト・ロケールの場合には、この文字数はバイト数として解釈する必要があります。

vi エディターはフルスクリーン・エディターなので、画面単位でテキストを編集できます。**vi** エディターは編集しているファイルのコピーを編集バッファに作成し、ファイルの内容を変更結果が保存されるまで変更しません。ディスプレイ画面上のカーソル位置は、ファイル内におけるカーソル位置を示し、サブコマンドはカーソル位置のファイルに影響を与えます。

vi エディターに関する制限

以下のリストは、**vi** エディターの最大制限値を示したものです。これらの値のカウントは、1 バイト文字を想定しています。

- グローバル・コマンド・リスト当たり 256 文字まで。
- 1 つのシェル・エスケープ・コマンドには 2048 文字まで。
- 1 つの文字列値オプションには 128 文字まで。
- 1 つのタグ名には 30 文字まで。

- 合計 2048 文字で 128 マップ・マクロ。
- 1,048,560 行と見なされる。
- マクロ名およびマクロ・テキストは 100 文字まで。

注: vi エディターは、最大 2 GB の編集バッファをサポートします。

vi 編集モード

vi エディターは、以下のモードで操作します。

項目	説明
コマンド・モード	vi エディターを始動すると、コマンド・モードになります。テキスト入力モードでしか指定できないものを除き、すべてのサブコマンドを入力できます。vi エディターは、サブコマンドおよび他のモードが終了すると、コマンド・モードに戻ります。サブコマンドを取り消すときには、Esc キーを押します。
テキスト入力モード	テキストを追加するときには、このモードで vi エディターを使用します。以下のサブコマンドのいずれかでテキスト入力モードに入ります。以下のサブコマンドのいずれかでテキスト入力モードに入ります。すなわち、 a サブコマンド、 A サブコマンド、 i サブコマンド、 I サブコマンド、 o サブコマンド、 O サブコマンド、 cx サブコマンド (<i>x</i> はサブコマンドの有効範囲を表します)、 C サブコマンド、 s サブコマンド、 S サブコマンド、 R サブコマンドです。これらのいずれかのサブコマンドを入力すれば、テキストを編集バッファに入力できます。コマンド・モードに戻るときには、正常終了の Esc キーを押すか、割り込み (Ctrl-C キー・シーケンス) を押し異常終了します。
最終行モード	接頭部: (コロン)、/ (スラッシュ)、? (疑問符)、!(感嘆符)、または !! (感嘆符 2 つ) が付いたサブコマンドは、画面最下部に表示される行に入力されたデータを読み取ります。最初の文字を入力すると、vi エディターではカーソルが画面最下部に移動するので、そこにコマンドの残りの文字を入力します。Enter キーを押すとサブコマンドは実行され、割り込み (Ctrl-C キー・シーケンス) を押すと取り消されます。感嘆符 !! を使用すると、カーソルは 2 つの感嘆符が入力されたときだけ移動します。接頭部: を使用して最終行モードを入力するとき、以下の文字をカウントを指定するコマンドの前で使用すると、vi エディターはその文字に特別な意味を持たせません。 % カーソル位置に関係なくすべての行 \$ 最終行 . 現在行 注: 上矢印キーおよび下矢印キーを使用して、最終行モード時のサブコマンドの履歴を見ることが出来ます。

vi エディターのカスタマイズ

vi エディターは、以下の方法でカスタマイズできます。

- vi エディター・オプションの設定
- マクロの定義
- キーのマッピング
- 省略形の設定

vi エディター・オプションの設定

以下のリストでは、**set** コマンドを使用して変更できる vi エディター・オプションについて説明します。各オプションのデフォルト設定は **off** です。これらのトグル・オプションのいずれかをオンにすると、オプションの前にワード **no** を入力して再びオフにすることができます。 **autowrite** vi オプションを中止したい場合は、**noaw** を入力します。この場合、**no** はオプションをオフにし、**aw** は **autowrite** オプションを指定します。

注: vi オプションを入力するときは、小括弧を使用しないでください。

vi オプション (省略形)

autoindent (ai)

説明

テキスト入力モード時に、**shiftwidth** オプションで指定したタブ・ストップ間隔を使用して、前行の字下げ位置まで自動的に字下げします。デフォルトは **noai** です。カーソルを前のタブ・ストップまで戻すには、**Ctrl-D** キー・シーケンスを押します。このオプションは、グローバル・コマンドには無効です。

autoprin (ap)

編集バッファを変更するコマンドの後に、現在行を表示します。デフォルトは **ap** です。このオプションは、単一の行に入力した一連のコマンドのうち、最後のコマンドにのみ適用され、グローバル・コマンドには無効です。

autowrite (aw)

編集バッファをファイルに自動的に書き込むのは、**:n** サブコマンド、**:ta** サブコマンド、**Ctrl-A**、**Ctrl -]**、および **Ctrl -T** キー・シーケンス、ならびに **!** サブコマンドの前で、前回の **write** サブコマンド以降に編集バッファが変更された場合です。デフォルトは **noaw** です。

backtags (bt)

Ctrl-T サブコマンドが、ファイル編集位置を前回の **Ctrl-]** サブコマンドが発行されたロケーションに戻すことができるようにします。**nobacktags** が設定されている場合、**Ctrl-T** は **Ctrl-]** と同じです。デフォルトは **backtags** です。

beautifying text (bf)

テキスト入力中に、編集バッファに制御文字 (タブ、改行、および用紙送りインジケータ) を入力できないようにします。デフォルトは **nobf** です。このオプションは、コマンド入力に適用されます。

closepunct (cp=)

特にテキストの折り返し時 (**wraptyp**e オプション) に終了句読点のリストを処理します。複数文字による句読点の前に文字数を付けます。例えば、**cp=3.;)** となります。vi コマンドは、折り返し時には終了句読点を分割しません。

directory (dir=)

編集バッファが入っているディレクトリを表示します。デフォルトは **dir = /var/tmp** です。

edcompatible (ed)

複数の置換を行う間にサブコマンド接尾部 **g** (グローバル) および **c** (確認) を保持し、**r** (読み取り) 接尾部を **r** サブコマンドと同様に機能させます。デフォルトは **noed** です。

exec (exec)

このオプションを設定しなければ、現在のディレクトリ名が **HOME** 環境変数で指定されていない限り、初期化時に現行ディレクトリ内の **.exec** ファイルがすべて無視されます。デフォルトは **noexec** です。

hardtabs (ht=)

vi エディターに対して、ディスプレイ画面上のハードウェア・タブ・ストップの間隔を指示します。(このオプションは、基礎となる端末または端末エミュレーターのタブ設定と一致する必要があります。) デフォルトは **ht=8** です。

history (hist=)

最終行モード時のコマンドの履歴を見る場合に使用される最大履歴数を設定します。初期値は **hist=32** です。**tvi** コマンドの場合、最大履歴数は **0 (hist=0)** です。

ignorecase (ic)

正規表現の検索中に大文字と小文字の違いを無視します。デフォルトは **noic** です。

linelimit (ll=)

-y コマンド・ライン・オプションに関して最大行数を設定します。このオプションは、**.exec** ファイルまたは **EXINIT** 環境変数とともに使用する場合のみ有効です。

lisp (lisp)

(), { }, [[, および]] の特殊な意味を除去し、s 式に対して = (フォーマット出力) 演算子を使用可能にして、リスト処理 (LISP) プログラムを編集できるようにします。デフォルトは **nolisp** です。

list (list)

タブ (^I) 付きテキストとマーク付きの行末 (\$) を表示します。デフォルトは **nolist** です。

magic (magic)

パターンを検索するときに、.(ピリオド)、[(左大括弧)、および * (アスタリスク) を特殊文字として扱います。オフ・モードでは、() (小括弧) と \$ (ドル記号) のみが特殊な意味を保持します。ただし、前に円記号 (¥) を付けると、他の文字の特殊な意味を復活させることができます。デフォルトは **magic** です。

vi オプション (省略形)	説明
mesg (mesg)	ビジュアル・モードで設定されている間、端末への書き込み許可をオンにします。このオプションは、 .exrc ファイルまたは EXINIT 環境変数とともに使用する場合のみ有効です。デフォルトは on です。
modeline (modeline)	このオプションをファイルの最初の 5 行または最後の 5 行に使用すると、vi エディターのコマンド・ラインが実行されます。vi エディターのコマンド・ラインは、1 行のどこにでも挿入できます。vi エディターにコマンド・ラインを認識させるには、その行のスペースまたはタブの後に文字列 ex: または vi: を付ける必要があります。コマンド・ラインは、2 つ目の : (コロン) で終わります。vi エディターは、1 つ目のコロンと 2 つ目のコロンの間にあるデータを、vi エディター・コマンドとして解釈しようとしません。デフォルトは nomodeline です。
novice	novice モードになっていることを示します。この値を set コマンドで変更することはできません。
number (nu)	各行に接頭部として行番号を付けて表示します。デフォルトは nonu です。
optimize (opt)	カーソル・アドレッシング機能がない端末の処理速度を高めめます。デフォルトは noopt です。
paragraphs (para=)	段落を開始する vi マクロ名を定義します。デフォルトは para=IPLPPPQPP¥ LIpplpipnbp です。 .P マクロのように単一文字の nroff マクロの場合、段落を再指定するときには引用文字としてスペースを指定する必要があります。
partialchar (pc=)	倍角文字が完全に表示されない場合に、最後の表示列に表示されます。デフォルト文字は - (負符号) です。
prompt	コマンド・モードで、: (コロン) を出力して新しい vi エディターの入力を求めるプロンプトを表示します。デフォルトは on です。
readonly (ro)	永久的な読み取り専用モードを設定します。デフォルトは noreadonly です。
redraw (redraw)	ダム・ワークステーション上でスマート・ワークステーションをシミュレートします。デフォルトは nore です。
remap	他のマクロに関するマクロを定義できるようにします。デフォルトは on です。
report (re=)	メッセージが表示される前にコマンドを繰り返すことができる回数を設定します。グローバル・サブコマンドのように、メッセージを多く生成するサブコマンドの場合は、コマンド・シーケンスが完了するとメッセージが表示されます。デフォルトは report=5 です。
scroll (scr=)	上下にスクロールする際のスクロール行数を設定します。デフォルトは、ウィンドウ・サイズの 1/2 に相当する行数から端数を切り捨てた行数です。
sections (sect=)	セクションを開始する vi マクロを定義します。デフォルトは sect=NHSHHH¥ HUuhsh+c です。 .P マクロのように単一文字の nroff マクロの場合、段落を再指定するときには引用文字としてスペースを指定する必要があります。
shell (sh=)	! サブコマンドまたは !: サブコマンドのシェルを定義します。デフォルトはログイン・シェルです。
shiftwidth (sw=)	autoindent オプション、シフト・コマンド (> と <)、およびテキスト入力コマンド (Ctrl-D と Ctrl-T キー・シーケンス) で使用するソフトウェア・タブ・ストップの間隔を設定します。vi オプションは、行頭の字下げにのみ影響を与えます。デフォルトは sw=8 です。
showmatch (sm)) (右小括弧) または] (右大括弧) を入力すると、それに対応している ((左小括弧) または [(左大括弧) を表示します。デフォルトは nosm です。
showmode (smd)	vi エディターが入力モードになっていることを示すメッセージを表示します。デフォルトは nosmd です。
slowopen (slow)	挿入時のディスプレイ画面の更新を延期します。デフォルトは noslow です。
tabstop (ts=)	表示されたファイル内でタブ・ストップの間隔を設定します。デフォルトは ts=8 です。
tags (tags =)	ctags コマンドを使用して作成された関数名データベース・ファイルの検索パスを定義します。デフォルトは tags=tags¥ /usr/lib/tags です。

vi オプション (省略形)	説明
term (term=)	使用中のワークステーションのタイプを設定します。デフォルトは term=\$TERM です。この場合、 \$TERM は TERM シェル変数の値です。vi エディターで短いフォーマットのメッセージを表示できるようにします。デフォルトは noterse です。
terse (terse)	
timeout (to)	文字入力の時間制限を 2 秒に設定します。この制限により、 timeout オプションを設定するときにマクロ内の文字を別個の文字として入力し、処理できます。マクロの使用を再開するには、 notimeout オプションを設定します。デフォルトは to です。
ttytype	使用する端末の tty タイプを示します。この値を vi エディターで変更することはできません。
warn (warn)	編集バッファで変更を行った後でシェル・コマンドを初めて発行するが、ファイルへの書き込みが済んでいない場合、! サブコマンドがシェル・コマンドを実行する前に警告メッセージを表示します。デフォルトは warn です。
window (wi=)	テキスト・ウィンドウ 1 つ分に表示される行数を設定します。600 ポー未満であれば 8 行、1200 ポーの場合は 16 行、1200 ポーを超える場合はフルスクリーンの行数より 1 行少ない行数です。
wrapmargin (wm=)	ある行から次の行までの自動ワード折り返しのマージンを設定します。デフォルトは wm=0 です。値 0 を指定すると、ワード折り返し機能がオフになります。
wrapscan (ws)	文字列検索を編集バッファの終わりから先頭に折り返せるようにします。デフォルトは ws です。
wraptype (wt=)	行末でワードを折り返すために使用するメソッドを示します。デフォルト値は general です。以下の 4 つの値から 1 つ指定できます。
	general 2 つの文字間のホワイト・スペースとしてワードの切れ目で折り返せるようにします。この設定はデフォルトです。
	word ワードのところで折り返せるようにします。
	rigid 列のところ (しかも、終了句読点の前) で折り返せるようにします。
	flexible 列のところを折り返せるようにしますが、1 文字の句読点はマージンを超えて延長できます。
writeany (wa)	通常は write サブコマンドの前に実行される検査をオフにします。デフォルトは nowa です。

デフォルト設定から変更された vi エディター設定のリストを表示するには、**set** と入力してスペース・バーを押します。コマンド・モードに戻るには、**Enter** キーを押します。

vi エディター設定の詳細リストを表示するには、**set all** と入力します。コマンド・モードに戻るには、**Enter** キーを押します。

vi エディターをオンにするには、**set Option** と入力します。このコマンドを実行すると、自動的にコマンド・モードに戻ります。

複数の vi エディター・オプションをオンにするには、**set Option Option Option** と入力します。このコマンドを実行すると、指定した 3 つの vi エディター・オプションがオンになり、コマンド・モードに戻ります。

vi エディター・オプションをオフにするには、**set noOption** と入力します。このコマンドを実行すると、自動的にコマンド・モードに戻ります。

vi エディター・オプションの値を変更するには、**setOption=Value** と入力します。このコマンドを実行すると、自動的にコマンド・モードに戻ります。

vi エディターの **:set** サブコマンドを使用すると、この編集セッション中にのみ使用するオプションを設定するか、またはこの編集セッションの将来のすべての編集セッションに使用するオプションを設定できます。

この編集セッションのみに使用する vi エディター・オプションを設定または変更するには、コマンド・ラインから **:set** サブコマンドを入力します。

すべての編集セッション用の vi オプションを設定するには、**:set** サブコマンドを **.profile** ファイル (ログイン時にシェルによって読み取られるファイル) 内の **EXINIT** 環境変数に入れるか、または **set** サブコマンドを **.exrc** ファイルに入れます。vi エディターは、まず **EXINIT** 環境変数を検索して、そのコマンドを実行します。**EXINIT** 環境変数が存在しなければ、vi エディターは次に **\$HOME/.exrc** ファイルを検索して、そのコマンドを実行します。最後に、以前の検索結果に関係なく、vi エディターはローカル **.exrc** ファイルを検索して、そのコマンドを実行します。

注: このプロセスは、**tvi** コマンド (トラステッド vi) 以外のコマンドに適用されます。この場合、vi エディターは **/etc/.exrc** ファイルのみを検索して実行します。

EXINIT 環境変数を設定してオプションを変更する方法については、**environment** ファイル内の環境変数の説明のセクションを参照してください。

.exrc ファイルには、以下のように **set Option=Value** フォーマットのサブコマンドを入れることができます。

```
set cp=3 . . . ;
```

.exrc ファイルにコメントを追加するには、行の先頭文字として " (二重引用符) を使用します。

マクロの定義

サブコマンドまたはサブコマンド・シーケンスを頻繁に使用する場合は、vi エディターを使用して、そのサブコマンドまたはシーケンスを発行するマクロを定義することができます。

マクロを定義するには、サブコマンドのシーケンスに英字の名前を付けてバッファーに入れます。a から z までの小文字を使用するとバッファーの内容がオーバーレイされ、A から Z までの大文字を使用すると、テキストはバッファーの以前の内容に追加されるので、マクロを個々に構築できます。

例えば、ワード **corner** を検索し、見つかった **corner** の後の 3 行目を現在の行にするバッファー・マクロ **c** を定義するには、次のコマンドを入力します。

```
o /corner/+3
```

次に、Esc キーを押して、次のコマンドを入力します。

```
"c
```

この場合、**c** はバッファー・マクロ名です。

定義済みバッファーの以前の内容にテキストを追加するには、**o viSubcommand** と入力し、Esc キーを押して、**"CapitalLetter** と入力します。この場合、**CapitalLetter** 変数は、A から Z までの大文字を指定します。例えば、**corner** という言葉を捜して、更にコマンドを追加できるバッファー・マクロ **T** を作成するには、次のように入力します。

```
o corner
```

次に、Esc キーを押して、次のコマンドを入力します。

"T

この場合、T はバッファ・マクロ名です。このプロセスを何度でも繰り返して、同じバッファにさらに vi サブコマンドを追加することができます。

例えば、カーソルを前の行に移動してその行を削除するコマンドを追加するには、次のコマンドを入力します。

o -dd

この場合、- (負符号) は、カーソルを 1 行上に移動することを意味し、dd は、現在行を削除することを意味します。Esc キーを押して次のコマンドを入力します。

"Tdd

マクロを開始するには、@Letter と入力します。この場合、Letter 変数は、使用したいバッファ・マクロの英字名を指定します。同じマクロを再度使用するには、@@ (アットマーク 2 つ) を入力します。例えば、@T と入力すると、T バッファ・マクロが開始され、**search**、**move cursor**、および **delete line** コマンドが実行されます。@@T と入力すると、T バッファ・マクロが再度開始されます。

システムで使用する文字セットは、照合テーブルで定義されます。このテーブルは、vi マクロのパフォーマンスに影響を与えます。

キーのマッピング

:map、**:map!**、および **:ab** サブコマンドを使用して、コマンドまたはコマンド・シーケンスにキー・ストロークをマップすることができます。**:map** サブコマンドは、コマンド・モードで使用します。**:map!** サブコマンドおよび **:ab** サブコマンドは、テキスト入力モードで使用します。どちらのモードでも、今回以降のすべての編集セッションに使用するキー、または今回の編集セッション専用のキーをマップできます。

将来のすべての編集セッションに使用するキーをマップするには、サブコマンドを **\$HOME/.exrc** ファイルに入れます。vi エディターを始動するたびに、このファイルが読み取られます。マッピングは、すべての編集セッションで有効になります。

現在の編集セッション専用 のキーをコマンド・モード からマップするには、vi エディター・セッション中にサブコマンドを開始します。現在の編集セッション専用のキーをテキスト入力モード からマップするには、vi エディター・セッション中にコマンド・ラインにサブコマンドを入力します。マッピングは、現在の編集セッション中にのみ有効です。

重要: IBM® 3161 ASCII ディスプレイ・ステーション、IBM 3163 ASCII ディスプレイ・ステーション、または IBM 3101 ASCII ディスプレイ・ステーションを使用する場合は、vi エディターのデフォルトのキー・マッピングを使用すると、データが失われることがあります。デフォルトのマッピングを表示するには、**:map** サブコマンドを発行します。Esc-J または Shift-J キー・シーケンスの場合は、特定の問題が発生します。これらのキー・シーケンスを使用すると、現在のカーソル位置からファイルの終わりまでのすべての情報が削除されます。この問題を避けるには、**.exrc** ファイルを使用してこのキー・シーケンスを変更します。

:map サブコマンド、**:map!** サブコマンド、および **:ab** サブコマンドを以下のように定義して使用します。

項目
:map

説明

マクロをコマンド・モードで定義します。 **:map** サブコマンドを使用すると、vi エディター内で単一のキーを押して、指定したコマンドまたはコマンド・シーケンスを実行できます。

コマンド・モードでキーをマップするには、編集バッファが空の状態 vi エディターを始動し、vi エディターが始動した後に **vi** コマンドで **vi** ファイル名を指定せず、バッファに何も入力しません。 **:map** サブコマンドを使用すると、以下の操作を行うことができます。

- 文字を編集コマンドのシーケンスにマップするには、次のように入力します。

```
:map Letter viSubcommand
```

- 以前にコマンド・モードでマップした文字をマップ解除するには、次のように入力します。

```
:unmap Letter
```

- コマンド・モード用の現在のマッピング・リストを表示するには、次のように入力します。

```
:map
```

以下のキーは、vi エディターでは使用しませんが、コマンド・モードで **:map** サブコマンドとともに使用することができます。

- 文字 g、K、q、V、および v
- 制御キー・シーケンス Ctrl-A、Ctrl-K、Ctrl-O、Ctrl-W、および Ctrl-X
- 記号 _ (下線)、* (アスタリスク)、円記号 (¥)、および = (等号)

既に vi エディターで使用しているキーをマップできますが、マップが有効になっている間は、そのキーの通常の機能を使用できなくなります。一部の端末では、コマンド・シーケンスをファンクション・キーにマップできます。LISP モードの場合は、vi エディターで使用されているので、= (等号) は使用できません。

ワード **map** が次に出現する位置を突き止めてワード **MAP** に変更するコマンド・シーケンスに、文字 **v** をマップするには、次のコマンドを入力します。

```
:map v /map<Ctrl-V><Enter>cwMAP<Ctrl-V><Esc><Ctrl-V><Enter>
```

上記の例は、vi エディターに対して、**map** が次に出現する位置を突き止め (**/map<Ctrl-V><Enter>**)、**map** を **MAP** に変更し (**cwMAP**)、ワード変更サブコマンドを終了し (**<Ctrl-V><Esc>**)、コマンドを入力します (**<Ctrl-V><Enter>**)。

要件: vi エディターで **Enter** キーが解釈されないようにするには、マップ時に先行の **Ctrl-V** キー・シーケンスを付ける必要があります。この条件は、**Esc**、**Backspace**、および **Delete** キーにも該当します。

制御文字 **Ctrl-A**、**Ctrl-K**、および **Ctrl-O** をマップするには、**Ctrl** キーと文字を同時に押します。例えば、ファイルを保存して連続する次のファイルを編集するコマンド・シーケンスに、**Ctrl-A** キー・シーケンスをマップするには、次のコマンドを入力します。

```
:map <Ctrl-A> :w<Ctrl-V><Enter>;n<Ctrl-V><Enter>
```

制御文字 **Ctrl-T**、**Ctrl-W**、および **Ctrl-X** をマップするには、**Ctrl-V** キー・シーケンスを使用して最初にエスケープする必要があります。

項目 説明

| (パイプ接続シンボル) をマップするには、次のように、2 つの Ctrl-V キー・シーケンスで最初にエスケープする必要があります。次の例では、シェルにエスケープし、ファイル `/etc/motd` を連結して出力を `wc` コマンドにパイプ接続するコマンド・シーケンスに、文字 `g` をマップします。

```
:map g :!cat /etc/motd <Ctrl-V><Ctrl-V>| wc<Ctrl-V><Enter>
```

端末でファンクション・キーをマップできる場合は、`#number` キー・シーケンスを使用して参照し、マップしたいファンクション・キーの番号を指定する必要があります。次の例では、1 ワードを削除してカーソルを下方に 3 ワード分だけ移動させるコマンド・シーケンスに、ファンクション・キー `F1` がマップされます。

```
:map #1 dwwww
```

ファンクション・キーのマッピングを機能させるには、端末タイプのファンクション・キーの出力が、`terminfo` ファイル内で定義した出力と一致する必要があります。これらの定義は `kfnumber` エントリで示されます。この場合、`kf1` はファンクション・キー `F1` を表し、`kf2` はファンクション・キー `F2` を表すという具合になります。ファンクション・キーを押したときに表示される出力が、このエントリと一致しない場合は、マッピングする前に、端末のセットアップ・モードを使用し、これらの端末データベース・エントリに合わせて設定を変更する必要があります。

また、`Home`、`End`、`Page Up`、および `Page Down` などの一定のキーボード特殊キーもマップすることができます。ほとんどの端末では、これらのキーは既に `vi` エディター内でマップされています。このマッピングは、`:map` サブコマンドを使用して確認できます。これらのキーがマップされていない場合は、以下のように `:map` サブコマンドを使用することができます。

```
:map <Ctrl-V><End> G
:map <Ctrl-V><Home> 1G
:map <Ctrl-V><PageUp> <Ctrl-F>
:map <Ctrl-V><PageDown> <Ctrl-B>
```

すべての現在のマップ・リストをコマンド・モードで表示させるには、`:map` サブコマンドを入力します。上記の例は、次のように表示されます。

```
v          v          /map<Ctrl-M>cwMAP<Ctrl-[>Ctrl-M>
<Ctrl-A>  <Ctrl-A>    :w<Ctrl-M>:n<Ctrl-M>
g          g          :!cat /etc/motd | wc <Ctrl-M>
```

ヒント: `Ctrl-V` および `Enter` キー・シーケンスは、`Ctrl-M` キー・シーケンスとして表示され、`Ctrl-V` および `Esc` キー・シーケンスは、`Ctrl-[` キー・シーケンスとして表示されます。

:map! テキスト入力モードで文字列を単一のキーにマップします。キーをテキスト入力モードでマップするには、編集バッファが空の状態では `vi` エディターを始動し、エディターが始動した後は、`vi` コマンドを使用して `vi` ファイル名を指定せず、バッファに何も入力しないでください。`:map!` サブコマンドを使用すると、以下の操作を行うことができます。

- テキスト入力モードで 1 文字を単一または複数の `vi` 文字列にマップするには、次のように入力します。

```
:map! Letter String
```

- 以前にテキスト入力モードでマップした文字をマップ解除するには、次のように入力します。

```
:unmap! Letter
```

- テキスト入力モードで特定のキーにマップされている既存の文字列のリストを表示するには、次のように入力します。

```
:map!
```

マップしたキーをテキスト入力モードで入力すると、指定した文字列が生成されます。`Ctrl-V` および `Esc` キー・シーケンスを使用すると、コマンド・モードになり、現在のワード (`bbw`) の先頭まで戻り、`cw` (ワード変更) サブコマンドが始動されます。例えば、次のように入力します。

```
:map! % <Ctrl-V><Esc>bbwcw
```

テキストを入力するときに、ワードを誤って入力したことに気付いたら、`%` (パーセント) キーを押し、そのワードを入力し直すと変更できます。自動的に挿入モードに戻ります。

重要: `:map!` サブコマンドに使用するキーは、慎重に選択してください。いったんキーをマップすると、最初に `:unmap!` サブコマンドを出さないでキーをテキストとして入力することはできません。

項目	説明
:ab	<p>テキスト入力モードで使用できるように、キーまたはキー・シーケンスを文字列にマップします。:ab サブコマンドは、複数の繰り返されるフレーズ、名前、またはタイトルを処理するテキストを入力するときに便利です。</p> <p>以下の例では、ワード <code>city</code> をテキスト入力モードで入力し、後にホワイト・スペース、ピリオド、またはコンマを付けると、フレーズ <code>Austin, Texas 78759</code> に置き換えられます。</p> <pre>:ab city Austin, Texas 78759</pre> <p>例えば、テキスト入力中に、次のように入力するものと仮定します。</p> <pre>My current residence is city.</pre> <p>Tab キーを押すと、ワード <code>city</code> が次のように展開されます。</p> <pre>My current residence is Austin, Texas 78759.</pre> <p>ワード内に含まれる省略形は、展開されません。例えば、<code>My current residence iscity</code> と入力しても、ワード <code>iscity</code> は展開されません。</p> <p>:map! サブコマンドを使用して挿入モードに省略形をマップすると、発生する位置にかかわらず、出現した省略形はすべて展開されます。前述の例 (<code>:map! city Austin, Texas 78759</code>) に :map! サブコマンドを使用すると、前後の文字に関係なく、<code>city</code> ワードをタイプした場合、常にそのワードは <code>Austin, Texas 78759</code> に展開されます。したがって、ワード <code>iscity</code> は <code>isAustin, Texas 78759</code> となります。</p> <p>重要: :ab サブコマンドに使用するキーは、慎重に選択してください。キーを定義すると、そのキーは最初に :unab サブコマンドを発行しなければテキストとして入力できなくなります。</p>

省略形の設定

set コマンドは、省略形が分離した語である場合にのみ、省略形を文字列に置き換えるという点を除けば、**set** コマンドは **map!** コマンドと同様に機能します。vi エディターの **set** コマンドを使用すると、以下の操作を行うことができます。

- 既存の省略形のリスト
- 省略形の除去
- 省略形の設定 (定義)

ヒント: 編集バッファが空の状態 vi エディター を始動してください。vi エディターが始動した場合は、**vi** コマンドを使用して vi ファイル名を指定したり、バッファに入力したりしないでください。Esc キーを押して、コマンド・モードになっていることを確認してください。

項目	説明
省略形のリスト	:ab コマンドを入力すると、既存の省略形がリストされます。Enter キーを押すと、コマンド・モードに戻ります。
省略形の除去	省略形を除去するには、 :anab <i>Abbreviation</i> コマンドを入力します。この場合、 <i>Abbreviation</i> 変数では、それ以上省略したくない文字列を指定します。

項目
省略形の設定 (定義)

説明
省略形を設定するには、**:ab Abbreviation String** コマンドを入力します。この場合、*Abbreviation* 変数では、省略形として定義する文字列を指定し、*String* 変数では省略する文字列を指定します。省略形が別個のワードである場合にのみ、省略形は文字列に置き換えられます。

例えば、テキスト入力モードで **:ab kn upper** コマンドを入力してから、**acknowledge** と入力すると、ワード **acknowledge** に含まれる文字列 **kn** は別個のワードではないので、省略形設定文字列は開始されません。

ただし、テキスト入力モードで **:ab kn upper** コマンドを入力してから **make the kn line all kncase** と入力すると、結果は **make the upper line all uppercase** となります。

フラグ

項目	説明
-cSubcommand	vi による表示を開始する前に、 ex エディター・サブコマンドを実行します。カーソルは、最後に実行されたサブコマンドの影響を受ける行に移動します。 -c ' ' の場合と同様に、 null オペランドを入力すると、 vi エディターによりカーソルがファイルの先頭行に移動します。 -c フラグは、 + フラグと一緒に使用できません。同時に両方のフラグを指定しないでください。
-l	LISP モードで vi エディターに入ります。このモードでは、LISP コードに適した字下げが vi エディターによって作成され、 () 、 { } 、 [[,]] などのサブコマンドは、LISP のコードに適した動作をするように変更されます。
-r [File]	vi エディター やシステムの誤動作が発生した後で、ファイルを回復します。 <i>File</i> 変数を指定しなければ、 vi エディターはすべての保管ファイルのリストを表示します。
-R	readonly オプションを設定し、ファイルの上書きを防止します。
-tTag	<i>Tag</i> 変数の入ったファイルを編集し、 vi エディターをその定義に配置します。このフラグを使用するには、ユーザーは最初に ctags コマンドを使用して関数名とその位置からなるデータベースを作成する必要があります。
-v	verbose モードで vi エディターに入ります。
-wNumber	デフォルト・ウィンドウ・サイズを、 <i>Number</i> 変数で指定した値に設定します。このフラグは低速回線で vi エディターを使用するときに便利です。
-yNumber	最大行数設定値 1,048,560 を 1024 よりも大きい任意の値で上書きします。 vi エディターがバッファ一操作で余分に行を使用するので必要な行数を 2 度要求する必要があります。
+ [Subcommand]	編集開始前に ex エディター・サブコマンドを実行します。 <i>Subcommand</i> 変数を指定しなければ、カーソルはファイルの先頭行に置かれます。この + フラグは、 -c フラグと一緒に使用できません。同時に両方のフラグを指定しないでください。

vi 一般サブコマンドの構文

サブコマンドを入力するには、以下の一般構文を使用します。

[Named_Buffer] [Operator] [Number] Object

ヒント: 大括弧はオプション項目を示します。

項目	説明
[Named_Buffer]	一時テキスト・ストレージ域を指定します。
[Operator]	サブコマンドやアクションを指定し、vi エディターを指示します。
[Number]	アクションまたは行アドレスの範囲を整数で指定します。
Object	実行対象として、テキスト・オブジェクト (文字、ワード、ステートメント、段落、セクション、文字列) またはテキスト位置 (行、現在行における位置、画面位置) を指定します。

サブコマンドの前のカウント

多くのサブコマンドの前には数値を指定することができます。vi エディターはこの数値を以下のいずれかの方法で解釈します。

- *Number* パラメーターで指定した行に移動する。
5G
10Z
- *Number* パラメーターで指定した桁に移動する。
25|
- *Number* パラメーターで指定した行数だけ上下にスクロールする。
10Ctrl-U
10Ctrl-D

vi エディターのサブコマンド

サブコマンドを使用すると、以下のようなアクションを実行できます。

- カーソルの移動
- テキストの編集
- ファイルの操作
- その他のアクション

カーソルの移動

サブコマンドを使用すると、次の方法によってファイル内でカーソルを移動できます。

- 1 行内の移動
- 文字位置による 1 行内の移動
- ワードへの移動
- 行位置による移動
- センテンス、段落、またはセクションへの移動
- 画面の再ドロウによる移動
- ページングとスクロール
- パターン検索
- ファイル内の特定位置にマーク付けして戻る

1 行内の移動

コマンド・モードで以下のサブコマンドを入力します。不完全なコマンドは、Esc キーを押すと取り消すことができます。vi サブコマンドのフォーマットについては、『vi 一般サブコマンドの構文』のセクションを参照してください。

項目	説明
左向き矢印、または h か Ctrl-H	カーソルを 1 文字左に移動します。
下矢印、 j 、または Ctrl-J か Ctrl-N	カーソルを 1 行下に移動します (桁位置は移動しません)。
上矢印、または k か Ctrl-P	カーソルを 1 行上に移動します (桁位置は移動しません)。
右向き矢印または l	カーソルを 1 文字右に移動します。

文字位置による 1 行内の移動

コマンド・モードで以下のサブコマンドを入力します。不完全なコマンドは、Esc キーを押すと取り消すことができます。vi サブコマンドのフォーマットについては、『vi 一般サブコマンドの構文』のセクションを参照してください。

項目	説明
^	カーソルを最初の非空白文字に移動します。
0	カーソルを行の先頭に移動します。
\$	カーソルを行の終わりに移動します。
fx	カーソルを次の <i>x</i> 文字に移動します。
Fx	カーソルを最後の <i>x</i> 文字に移動します。
tx	カーソルを、次の <i>x</i> 文字の 1 桁前に移動します。
Tx	カーソルを、最後の <i>x</i> 文字の 1 桁後に移動します。
;	最後の f 、 F 、 t 、または T サブコマンドを繰り返します。
,	最後の f 、 F 、 t 、または T サブコマンドを反対方向に繰り返します。
Numberl	カーソルを指定した桁に移動します。

ワードへの移動

コマンド・モードで以下のサブコマンドを入力します。vi サブコマンドのフォーマットについては、『vi 一般サブコマンドの構文』のセクションを参照してください。

項目	説明
w	カーソルを次の small ワードに移動します。
b	カーソルを直前の small ワードに移動します。
e	カーソルを次の small ワードの最後に移動します。
W	カーソルを次の big ワードに移動します。
B	カーソルを直前の big ワードに移動します。
E	カーソルを次の big ワードの最後に移動します。

行位置による移動

コマンド・モードで以下のサブコマンドを入力します。vi サブコマンドのフォーマットについては、『vi 一般サブコマンドの構文』のセクションを参照してください。

項目	説明
H	カーソルを画面の一番上の行に移動します。
L	カーソルを画面の一番下の行に移動します。
M	カーソルを画面の中央の行に移動します。
+	カーソルを次行の最初の空白以外の文字に移動します。
-	カーソルを直前の行の最初の空白以外の文字に移動します。
Enter	カーソルを次行の最初の空白以外の文字に移動します。

センテンス、段落、またはセクションへの移動

コマンド・モードで以下のサブコマンドを入力します。不完全なサブコマンドは、Esc キーを押すと取り消すことができます。vi サブコマンドのフォーマットについては、『vi 一般サブコマンドの構文』のセクションを参照してください。

項目	説明
(カーソルを直前のセンテンスの先頭か、LISP モードのときは直前の s 式に移動します。
)	カーソルを次のセンテンスの先頭か、LISP モードのときは次の s 式に移動します。
{	カーソルを直前の段落の先頭か、LISP モードのときは次のリストに移動します。
}	カーソルを次の段落の先頭か、C モードのときは次のセクションに、または LISP モードのときは次のリストに移動します。
	カーソルを次のセクションか、LISP モードのときは次の関数に移動します。
	カーソルを直前のセクションか、LISP モードのときは直前の関数に移動します。

画面の再ドロワーによる移動

コマンド・モードで以下のサブコマンドを入力します。不完全なサブコマンドは、Esc キーを押すと取り消すことができます。vi サブコマンドのフォーマットについては、『vi 一般サブコマンドの構文』のセクションを参照してください。

項目	説明
z	画面最上部に現在行を移動して画面を再表示します。
z-	画面最下部に現在行を移動して画面を再表示します。
z.	画面中央に現在行を移動して画面を再表示します。
 Pattern/z-	<i>Pattern</i> パラメーターで指定した文字列を含む行を画面最下部に移動して画面を再表示します。

ページングとスクロール

コマンド・モードで以下のサブコマンドを入力します。不完全なサブコマンドは、Esc キーを押すと取り消すことができます。vi サブコマンドのフォーマットについては、『vi 一般サブコマンドの構文』のセクションを参照してください。

項目	説明
Ctrl-U	半画面上にスクロールします。
Ctrl-D	半画面下にスクロールします。
Ctrl-F	1 画面下にスクロールします。
Ctrl-B	1 画面上にスクロールします。
Ctrl-E	ウィンドウを 1 行下にスクロールします。
Ctrl-Y	ウィンドウを 1 行上にスクロールします。
z+	前ページ
z^	次ページ

パターン検索

コマンド・モードで以下のサブコマンドを入力します。不完全なサブコマンドは、Esc キーを押すと取り消すことができます。vi サブコマンドのフォーマットについては、『vi 一般サブコマンドの構文』のセクションを参照してください。

項目	説明
[<i>Number</i>] G	<i>Number</i> パラメーターで指定した行番号にカーソルを移動します。 <i>Number</i> パラメーターが指定されていない場合は、最終行にカーソルを移動します。
<i>I</i> Pattern	<i>Pattern</i> パラメーターで指定した文字列のある行の次の行にカーソルを置きます。
? <i>Pattern</i>	<i>Pattern</i> パラメーターで指定した文字列のある行の直前の行にカーソルを置きます。
n	<i>Pattern</i> パラメーターで指定したテキストの最後の検索を同方向に繰り返します。
N	<i>Pattern</i> パラメーターで指定したテキストの最後の検索を反対方向に繰り返します。
<i>I</i> Pattern + <i>Number</i>	<i>Pattern</i> パラメーターで指定した文字列と一致する行の後の指定した行番号にカーソルを置きます。
? <i>Pattern</i> ?- <i>Number</i>	<i>Pattern</i> パラメーターで指定した文字列と一致する行の前の指定した行番号にカーソルを置きます。
%	現在のカーソル位置にあるものと一致する括弧または中括弧を検索します。

テキストの編集

編集用のサブコマンドで、次の作業を実行できます。

- ファイル内の特定位置にマーク付けて戻る
- ファイルへのテキスト追加
- 入力モードでのテキスト変更
- コマンド・モードでのテキスト変更
- テキストのコピーと移動
- 変更結果の復元と繰り返し

ファイル内の特定位置にマーク付けて戻る

コマンド・モードで以下のサブコマンドを入力します。不完全なサブコマンドは、Esc キーを押すと取り消すことができます。vi サブコマンドのフォーマットについては、『vi 一般サブコマンドの構文』のセクションを参照してください。

項目	説明
"	カーソルを現在行の直前の位置に移動します。
"	カーソルを現在行の直前の位置を含む行の先頭に移動します。
mx	<i>x</i> パラメーターで指定した文字で現在位置をマークします。
` <i>x</i>	<i>x</i> パラメーターで指定したマークの位置にカーソルを移動します。
' <i>x</i>	カーソルを <i>x</i> パラメーターで指定したマークを含む行の先頭に移動します。

ファイルへのテキストの追加 (テキスト入力モード)

vi エディターをテキスト入力モードに変更するには、コマンド・モードで以下のサブコマンドを入力します。vi サブコマンドのフォーマットについては、『vi 一般サブコマンドの構文』のセクションを参照してください。

項目	説明
a <i>Text</i>	<i>Text</i> パラメーターで指定したテキストをカーソルの後に挿入します。テキスト入力モードは Esc キーを押すと終了します。
A <i>Text</i>	<i>Text</i> パラメーターで指定したテキストを行の最後に追加します。テキスト入力モードは Esc キーを押すと終了します。
i <i>Text</i>	<i>Text</i> パラメーターで指定したテキストをカーソルの前に挿入します。テキスト入力モードは Esc キーを押すと終了します。
I <i>Text</i>	<i>Text</i> パラメーターで指定したテキストを行の最初の非空白文字の前に挿入します。テキスト入力モードは Esc キーを押すと終了します。
o	空行を現在行の下に追加します。テキスト入力モードは Esc キーを押すと終了します。

項目	説明
O	空行を現在行の上に追加します。テキスト入力モードは Esc キーを押すと終了します。

入力モードでのテキスト変更

テキスト入力モードでのみ以下のサブコマンドを使用してください。これらのコマンドは、コマンド・モードでは別の意味を持っています。vi サブコマンドのフォーマットについては、『vi 一般サブコマンドの構文』のセクションを参照してください。

項目	説明
Ctrl-D	前の自動字下げストップに戻ります。
^ Ctrl-D	現在行についてのみ自動字下げ処理を終了します。
0Ctrl-D	カーソルを左マージンに戻します。
Esc	挿入を終了し、コマンド状態に戻ります。
Ctrl-H	最後の文字を消去します。
Ctrl-Q	xon が使用不可になっている場合に任意の文字を入力します。
Ctrl-V	任意の文字を入力します。
Ctrl-W	最後の small ワードを消去します。
¥	消去文字と抹消文字を引用します。
Ctrl-?	挿入または Ctrl-D キー・シーケンスに割り込んで終了します。

コマンド・モードでのテキスト変更

コマンド・モードで以下のサブコマンドを使用します。不完全なサブコマンドは、Esc キーを押すと取り消すことができます。vi サブコマンドのフォーマットについては、『vi 一般サブコマンドの構文』のセクションを参照してください。

項目	説明
C	行の残りの部分を変更します (c\$ と同じ)。
cc	行を変更します。
cw	ワードを変更します。
cwText	<i>Text</i> パラメーターで指定したテキストにワードを変更します。
D	行の残りの部分を削除します (d\$ と同じ)。
dd	行を削除します。
dw	ワードを削除します。
J	行を結合します。
rx	現行文字を、 <i>x</i> で指定した文字に置き換えます。 <i>Text</i> パラメーターで指定したテキストに文字を上書きします。
RText	文字を置き換えます (cl と同じ)。
s	行を置き換えます (cc と同じ)。
S	前の変更を元に戻します。
u	カーソル位置の文字を削除します。
x	カーソル位置の前の文字を削除します (dh と同じ)。
<<	1 行左にシフトします。
<L	カーソルから画面の終わりまでの全行を左にシフトします。
>>	1 行右にシフトします。
>L	カーソルから画面の終わりまでの全行を右にシフトします。
~	カーソル位置の文字が大文字であれば小文字に、小文字であれば大文字に変更します。
!	LISP 用に字下げします。

テキストのコピーと移動

コマンド・モードで以下のサブコマンドを使用します。不完全なサブコマンドは、Esc キーを押すと取り消すことができます。vi サブコマンドのフォーマットについては、『vi 一般サブコマンドの構文』のセクションを参照してください。

項目	説明
p	アンドゥ・バッファからテキストを戻し、カーソルの後に挿入します。
P	アンドゥ・バッファからテキストを戻し、カーソルの前に挿入します。
"xp	x バッファからテキストを戻します。
"xd	テキストを削除して x バッファに入れます。
y	後に続くオブジェクト (例えばワードであれば w) をアンドゥ・バッファに入れます。
"xy	後に続くオブジェクトを x バッファに入れます。x はどの文字でもかまいません。
Y	行をアンドゥ・バッファに入れます。

変更結果の復元と繰り返し

コマンド・モードで以下のサブコマンドを使用します。不完全なサブコマンドは、Esc キーを押すと取り消すことができます。vi サブコマンドのフォーマットについては、『vi 一般サブコマンドの構文』のセクションを参照してください。

項目	説明
u	最後の変更を元に戻します。
	ヒント: 元に戻した後は、カーソルは更新された現在行上の先頭の子非空白文字に移動します。
U	最後の変更以後カーソルが行を残していなければ、現在行を復元します。
.	最後の変更を繰り返すか、" np " コマンドを増分します。

注:

- このサブコマンドは、元に戻された場合も含めて、最後の変更を繰り返します。したがって、元に戻された場合は、最後の変更が繰り返されるのではなく元に戻されたものが繰り返されます。
- このサブコマンドは、マクロに使用するものではありません。マクロを繰り返すには、@@ (アットマーク 2 つ) を入力します。

"n p 最後から n 番目に削除された行をすべて、またはブロック単位で検索します。

ファイルの操作

ファイル操作のサブコマンドを使用すると、以下のセクションで説明するタスクを実行することができます。

- ファイルへの変更保存
- 第 2 のファイルの編集
- ファイル・リストの編集
- ファイル情報の検索

ファイルへの変更保存

コマンド・モードで以下のサブコマンドを使用します。vi サブコマンドのフォーマットについては、『vi 一般サブコマンドの構文』のセクションを参照してください。

項目	説明
<code>:w</code>	編集バッファの内容を元のファイルに書き込みます。 ex エディター内でこのサブコマンドを使用している場合は、 <code>:</code> (コロン) は不要です。
<code>:w File</code>	<i>File</i> パラメーターで指定したファイルに編集バッファの内容を書き込みます。 ex エディター内でこのサブコマンドを使用している場合は、 <code>:</code> (コロン) は不要です。
<code>:w! File</code>	<i>File</i> パラメーターで指定したファイルに編集バッファの内容を上書きします。 ex エディター内でこのサブコマンドを使用している場合は、 <code>:</code> (コロン) は不要です。

第 2 のファイルの編集

コマンド・モードで以下のサブコマンドを入力します。 `vi` サブコマンドのフォーマットについては、『`vi` 一般サブコマンドの構文』のセクションを参照してください。

項目	説明
<code>:e File</code>	指定ファイルを編集します。 ex エディターからこのサブコマンドを使用している場合、 <code>:</code> (コロン) は不要です。
<code>:e!</code>	現行ファイルを再度編集し、すべての変更を破棄します。
<code>:e + File</code>	指定ファイルをファイルの終わりから編集します。
<code>:e + Number File</code>	指定ファイルを指定行番号から編集します。
<code>:e #</code>	代替ファイルを編集します。通常は、 <code>:e</code> コマンドで別のファイルにアクセスする前のファイル名が代替ファイルです。ただし、新しいファイルを呼び出したときに現行ファイル内で変更が保留になっている場合には、新しいファイルが代替ファイルになります。これは、 Ctrl-A サブコマンドと同じです。
<code>:r File</code>	新しい行を現行行の下に追加することによって、ファイルを編集バッファに読み取ります。 ex エディターからこのサブコマンドを使用している場合、 <code>:</code> (コロン) は不要です。
<code>:r !Command</code>	指定したコマンドを実行し、新しい行を現在のカーソル位置に追加することによってファイル内に出力を入れます。
<code>:ta Tag</code>	<i>Tag</i> タグの入っているファイルを、そのタグ位置から編集します。このサブコマンドを使用するには、最初に ctags コマンドを使用して関数名とそれらの位置のデータベースを作成する必要があります。ex エディターからこのサブコマンドを使用している場合、 <code>:</code> (コロン) は不要です。
Ctrl-]	現行のワードと関連したタグの入っているファイルを、そのタグ位置から編集します。このサブコマンドを使用するには、最初に ctags コマンドを使用して関数名とそれらの位置のデータベースを作成する必要があります。 Ctrl-T は、前回の Ctrl-] サブコマンドが発行された編集位置でファイルを編集します。複数の Ctrl-] サブコマンドが発行されている場合は、 Ctrl-] サブコマンドが発行された前回の編集位置に戻るのに、複数の Ctrl-T サブコマンドを使用することができます。
Ctrl-A	代替ファイルを編集します。代替ファイルは、通常は前の現行ファイル名です。ただし、新しいファイルを呼び出したときに現行ファイル内で変更が保留になっている場合には、新しいファイルが代替ファイルになります。このサブコマンドは <code>:e #</code> サブコマンドと同じです。

ファイル・リストの編集

コマンド・モードで以下のサブコマンドを入力します。 `vi` サブコマンドのフォーマットについては、『`vi` 一般サブコマンドの構文』のセクションを参照してください。

項目	説明
<code>:n</code>	コマンド・ラインで入力したリスト内の次のファイルを編集します。 ex エディターからこのサブコマンドを使用している場合は、 <code>:</code> (コロン) は不要です。
<code>:n Files</code>	編集する新しいファイル・リストを指定します。 ex エディターからこのサブコマンドを使用している場合は、 <code>:</code> (コロン) は不要です。

ファイル情報の検索

コマンド・モードで以下のサブコマンドを入力します。 `vi` サブコマンドのフォーマットについては、『`vi` 一般サブコマンドの構文』のセクションを参照してください。

項目	説明
Ctrl-G	現行ファイル名、現在行番号、ファイル内の行数、ファイル内におけるカーソル位置をパーセントで表示します。

その他のアクション

vi エディターには、以下のセクションで説明するサブコマンドがあります。

- スクリーン調整
- シェル・コマンドの入力
- vi エディターへの割り込みと終了

スクリーン調整

コマンド・モードで以下のサブコマンドを入力します。不完全なサブコマンドは、Esc キーを押すと取り消すことができます。vi サブコマンドのフォーマットについては、『vi 一般サブコマンドの構文』のセクションを参照してください。

項目	説明
Ctrl-L	画面を消去し、再表示します。
Ctrl-R	画面を再表示し、@ (アットマーク) でマークされたブランク行を削除します。
zNumber	ウィンドウの長さを指定した行数にします。

シェル・コマンドの入力

以下のサブコマンドを使用すると、vi エディターでコマンド内でコマンドを実行することができます。これらのサブコマンドは、コマンド・モードで入力します。vi サブコマンドのフォーマットについては、『vi 一般サブコマンドの構文』のセクションを参照してください。

項目	説明
:sh	複数のコマンドを実行できるように、シェルに入ります。Ctrl-D キー・シーケンスを押すと、vi エディターに戻ることができます。ex エディター内でこのサブコマンドを使用している場合は、: (コロン) は不要です。
!:Command	指定したコマンドを実行してから vi エディターに戻ります。ex エディター内でこのサブコマンドを使用している場合は、: (コロン) は不要です。
!!	ヒント: !! サブコマンドの後の特殊文字 # (代替ファイル)、% (現行ファイル)、および ! (直前のコマンド) は展開されます。これらの文字が展開されないようにするには、円記号 (¥) を使用してください。
Number!!Command	最後の !:Command サブコマンドを繰り返します。指定したコマンドを実行し、Number で指定した行をコマンドの出力に置き換えます。数値を指定しなかった場合、デフォルト値は 1 となります。コマンドが標準入力を求める場合、指定した行は入力として使用されます。
!Object Command	指定したコマンドを実行し、Object パラメーターで指定したオブジェクトをコマンドの出力に置き換えます。コマンドが標準入力を求める場合、指定したオブジェクトは入力として使用されます。

vi エディターへの割り込みと終了

コマンド・モードで以下のサブコマンドを入力します。vi サブコマンドのフォーマットについては、『vi 一般サブコマンドの構文』のセクションを参照してください。

項目	説明
Q	コマンド・モードで ex エディターに入ります。
ZZ	vi エディターを終了して変更を保存します。
:q	vi エディターを終了します。編集バッファの内容を変更した場合は、 vi エディターは警告メッセージを表示し、終了しません。 ex エディターからこのサブコマンドを使用している場合は、 : (コロン) は不要です。
:q!	vi エディターを終了します。編集バッファは破棄されます。 ex エディターからこのサブコマンドを使用している場合は、 : (コロン) は不要です。
Esc	テキスト入力または不完全なサブコマンドを終了します。
Ctrl-?	サブコマンドに割り込みます。

終了状況

以下の終了値が戻されます。

項目	説明
0	正常に完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。

入力ファイル

入力ファイルは必ずテキスト・ファイルか、同様のファイルである必要があります。ただし、ファイルの最後の行が不完全であってはなりません (つまり、ヌル文字が含まれていないことが必要です)。

.exrc ファイルは、**ex** コマンドが記述されたテキスト・ファイルである必要があります。

\$HOME/.vi_history ファイルは、自動生成されるテキスト・ファイルで、最終行モード時のコマンドの履歴を記録します。

デフォルトでは、**vi** エディターはファイルから編集する行を読み取る際に、各行を **vi** エディター・コマンドとして解釈しません。

関連情報:

ctags コマンド

ex コマンド

tvi コマンド

.profile ファイル

view コマンド

目的

vi エディターを読み取り専用モードで始動します。

構文

```
view [ -cSubcommand ] [ -l ] [ -t Tag ] [ -wNumber ] [ -y ] [ -r [ File ] ] [ + [ Subcommand ] ] [ File ... ]
```

説明

view コマンドは、**vi** フルスクリーン・エディターを読み取り専用モードで始動します。読み取り専用モードは、誤ってファイルを変更するのを避けるのに役立ちます。読み取り専用モードをオーバーライドする

には、コマンドの実行時に !(感嘆符) を使用します。 *File* パラメーターには表示したいファイルの名前を指定します。ファイル内での移動には、**vi** サブコマンドを使用します。**view** コマンドを終了するには、**:q** サブコマンドを使用してください。ファイルを変更する場合は、Esc キーおよび **wq!** を押すと変更内容を保管することができます。

フラグ

項目	説明
-c <i>Subcommand</i>	vi による表示を開始する前に、 ex エディター・サブコマンドを実行します。 -c '' のように null オペランドを入力すると、エディターはカーソルをファイルの最終行に移動します。
-l	LISP 言語でプログラムを書くために設計された特殊機能を備えたバージョンの vi エディターに入ります。このモードでは、 vi エディターは LISP プログラミング用に適宜字下げされ、 (、) 、 { 、 } 、 [[および]] などのサブコマンドは、LISP で正しく動作するように修正されます。
-r [<i>File</i>]	エディターかシステムがクラッシュした後でファイルを回復させます。 <i>File</i> パラメーターを指定しない場合には、エディターはすべての保管ファイルのリストを表示します。
-t <i>Tag</i>	<i>Tag</i> パラメーターで指定したタグの入っているファイルを編集し、エディターをその定義に配置します。このフラグを使用するには、最初に ctags コマンドを使用して関数名とそれらの位置のデータベースを作成する必要があります。
-w <i>Number</i>	デフォルトのウィンドウ・サイズを <i>Number</i> パラメーターに指定した値に設定します。ユーザーの端末と低速の通信回線でエディターを実行しているシステムで通信するときに役立ちます。
-y	最大行数設定値 1,048,560 を 1024 よりも大きい任意の値で上書きします。
+ [<i>Subcommand</i>]	vi による表示を開始する前に、 <i>Subcommand</i> パラメーターで指定した ex エディター・サブコマンドを実行します。サブコマンドを指定しない場合は、ファイルの最終行にカーソルが移動します。

関連資料:

14 ページの『**vi** または **vedit** コマンド』

関連情報:

ctags コマンド

vmh コマンド

目的

MH コマンドで使用するためのビジュアル・インターフェースを開始します。

構文

```
vmh [ -prompt String ] [ -vmhproc CommandString | -novmhproc ]
```

説明

vmh コマンドは **MH** コマンドで使用するビジュアル・インターフェースを始動します。**vmh** コマンドは **MH** ウィンドウ管理プロトコルのサーバー側をインプリメントし、プロトコルのクライアント側をインプリメントするすべてのプログラムとの分割画面インターフェースを維持します。

vmh コマンドはコマンドの入力を要求し、それらをプロトコルのクライアント側に送信します。コマンドによって出力データが複数画面に分割されたウィンドウが作成された場合には、**vmh** コマンドはユーザーにサブコマンドの入力を要求します。**vmh** サブコマンドを入力すると、コマンド出力の特定部分を表示することができます。

vmh サブコマンド

項目	説明
Ctrl-L	画面をリフレッシュします。
スペース	次画面に進みます。
[Number] Enter	指定した行数だけ先に進みます。デフォルトは 1 行です。
[Number] d	指定値の 10 倍の行数だけ先に進みます。Number 変数のデフォルトは 1 で、移動行数は 10 行になります。
[Number] g	指定行に移動します。
[Number] G	ウィンドウの終わりに移動します。Number 変数を指定すると、コマンドは g フラグのように動作します。
[Number] u	指定値の 10 倍の行数だけ戻ります。Number 変数のデフォルトは 1 で、移動行数は 10 行になります。
[Number] y	指定行数だけ戻ります。デフォルトは 1 行です。
h	ヘルプ・メッセージを表示します。
q	出力を終了します。

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)、バージョン情報をリストします。 注: MH の場合、このフラグの名前のスペルは完全に入力しなければなりません。
-novmproc	ウィンドウ管理プロトコルなしでデフォルトの vmproc ディレクトリーを実行します。
-prompt <i>String</i>	指定した文字列をプロンプトとして使用します。
-vmhproc <i>CommandString</i>	ウィンドウ管理プロトコルのクライアント側を実現するプログラムを指定します。 デフォルトは msh プログラムです。

プロファイル・エントリー

以下のエントリーが、*UserMhDirectory/.mh_profile* ファイルに入力されます。

項目	説明
Path:	ユーザーの MH ディレクトリーを指定します。
mshproc:	MH シェルで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/vmh	vmh コマンドが入っています。

関連情報:

msh コマンド
mh_alias コマンド
mh_profile コマンド

vmo コマンド

目的

仮想メモリー・マネージャー (Virtual Memory Manager) の Tunable (チューナブル) パラメーターを管理します。

構文

vmo [**-p** | **-r**] [**-y**] { **-o** *Tunable* [= *Newvalue*] }

vmo [**-p** | **-r**] [**-y**] { **-d** *Tunable* }

vmo [**-p** | **-r**] [**-y**] **-D**

vmo [**-p** | **-r**] [**-F**] **-a**

vmo **-h** [*Tunable*]

vmo [**-F**] **-L** [*Tunable*]

vmo [**-F**] **-x** [*Tunable*]

注: 複数の **-o**, **-d**, **-x** および **-L** を指定できます。

説明

注: **vmo** コマンドは **root** からのみ実行できます。 **vmo** コマンドは、自己文書化コマンドです。一部のフラグまたはチューナブル・パラメーターに関する情報が、欠落していたり、期限切れになっている場合があります。 **-h** フラグ、 **-L** フラグ、または **-x** フラグを使用すると、すべてのフラグおよびチューナブル・パラメーターの最新リストを検索できます。

vmo コマンドを使用して、仮想メモリー・マネージャー (VMM) のチューニング・パラメーターを構成します。このコマンドは、すべての仮想メモリー・マネージャー (VMM) のチューニング・パラメーターの現行またはリブート値を設定したり、表示します。また、永続的な変更を行ったり、変更を次のリブートまで延ばすこともできます。コマンドがパラメーターを設定するのを表示するのかは、一緒に指定するフラグによって決まります。 **-o** フラグはこの両方のアクションを指定します。パラメーターの値を表示することも、パラメーターに新規の値を設定することもできます。

仮想メモリー・マネージャー (VMM) は、フリーの実メモリー・ページ・フレームのリストを保守します。ページ・フレームは、ページ・フォールトを満たすのに必要な仮想メモリー・ページを保持するために使用できます。フリー・リスト上のページ数が、 **minfree** パラメーターで指定した値を下回ると、VMM は、フリー・リストへ追加するためにページをスチールし始めます。VMM は、フリー・リストが、少なくとも **maxfree** パラメーターで指定したページ数になるまでページのスチールを続けます。

メモリー内のファイル・ページ (永続ページ) の数が、 **minperm%** パラメーターで指定した数を下回る場合、VMM は、 **repage** 率に関係なく、計算ページまたはファイル・ページのいずれかからフレームをスチールします。ファイル・ページの数、 **maxperm%** パラメーターで指定した数を上回る場合、VMM は、ファイル・ページからのみフレームをスチールします。この 2 つの間で、VMM は通常、ファイル・ページのみをスチールしますが、ファイル・ページの **repage** 率が計算ページの **repage** 率を上回る場合は、計算ページもスチールされます。

システムでページング・スペースがなくなる時点を判別するのに使用するしきい値を変更することもできます。 **npswarn** パラメーターは、ページング・スペースが少なくなってきた場合に、システムが警告プロセスを開始する時点で使用可能なページング・スペースのページ数を指定します。 **npskill** パラメーターは、システムがページング・スペースを解放するためにプロセスの停止を開始する時点で使用可能なページング・スペースのページ数を指定します。

注: オプション **-o**、**-d**、および **-D** は、仮想メモリー・マネージャーのチューナブル・パラメーターの値の変更を試みるもので、ワークロード・パーティション内ではサポートされません。

チューナブル・パラメーターの変更による影響について

このコマンドを誤って使用すると、パフォーマンスの低下またはオペレーティング・システムの障害の原因となります。**vmo** コマンドを試す前に、仮想メモリー・マネージャーのパフォーマンスの概要と **maxclient** パラメーターを使用した拡張 JFS ファイルシステムのキャッシュ制限の両方を熟読する必要があります。

チューナブル・パラメーターを変更する前に、まず、後述の チューナブル・パラメーターのセクションですべての特性について注意深く読み、その目的を十分に理解するために、すべての「参照」ポインターをたどるようにしてください。

次に、このパラメーターについての『診断とチューニング』のセクションがご使用のシステムの状況に実際に当てはまるかどうか、また、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を改善するのに役立つかどうかを確認する必要があります。

「診断」および「チューニング」のセクションがいずれも「該当せず」となっている場合は、AIX 開発担当者から特に指示がない限り、そのパラメーターは変更しないでください。

フラグ

項目	説明
-a	すべてのチューナブル・パラメーターの現行値、リポート値 (-r オプションを指定した場合) または永続値 (-p オプションを指定した場合) を、 <i>Tunable = Value</i> の対にして、1 行に 1 つずつ表示します。永続オプションの場合、リポート値と現行値が等しいときのみ、1 つのパラメーターに 1 つの値が表示されます。そうでない場合は、 NONE が値として表示されます。
-d Tunable	<i>Tunable</i> パラメーターをそのデフォルト値にリセットします。 <i>Tunable</i> パラメーターは、デフォルト値に設定されていないために変更が必要ですが、以下の基準の組み合わせの 1 つ以上を満たした場合、警告メッセージが表示され、パラメーターに対する変更は行われません。 <ul style="list-style-type: none">チューナブル・パラメーターのタイプは、Bosboot または Reboot です。チューナブル・パラメーターは Incremental タイプで、そのデフォルト値から変更されており、-r フラグと組み合わせて使用されることはありません。
-D	すべての <i>Tunable</i> 値を、それぞれのデフォルト値にリセットします。デフォルト値に設定されていないために変更が必要な <i>Tunable</i> が以下の基準のセットの 1 つ以上を満たしている場合、警告メッセージが表示され、変更は行われません。 <ul style="list-style-type: none">チューナブルのタイプは Bosboot または Reboot です。チューナブルは、Incremental タイプで、そのデフォルト値から変更されており、-r と組み合わせて使用されることはありません。
-F	すべてのチューナブルをリストするために -a 、 -L あるいは -x オプションがコマンド・ラインでのみ指定されている場合、制限付きチューナブル・パラメーターを強制的に表示させます。 -F フラグが指定されていない場合、制限付きチューナブルは、 display オプションで明確に指名されていない限り、表示されません。
-h [Tunable]	チューナブル・パラメーターを指定すると、 <i>Tunable</i> パラメーターに関するヘルプを表示します。指定しなければ、 vmo コマンド使用法ステートメントが表示されます。

項目

-L [Tunable]

説明

1 つまたはすべての Tunable の特性を、以下の形式で、1 行に 1 つずつリストします。

NAME	CUR	DEF	BOOT	MIN	MAX	UNIT	TYPE

memory_frames	128K		128K			4KB pages	S

maxfree	1088	1088	130	16	200K	4KB pages	D
minfree							
memory_frames							

minfree	960	960	122	8	200K	4KB pages	D
maxfree							
memory_frames							

...

ここで、

CUR = 現行値
 DEF = デフォルト値
 BOOT = リポート値
 MIN = 最小値
 MAX = 最大値
 UNIT =
 チューナブル計測単位
 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。チューナブルを変更する必要があります (指定された値が現行値と異なる)、**Bosboot** または **Reboot** のタイプである場合、あるいは **Incremental** のタイプで、**-r** を組み合わせて使用しない場合は、チューナブル値は変更されず、警告が表示されます。

新しい値を指定せずに、**-r** フラグを組み合わせて使用する場合は、Tunable のリポート値が表示されます。新しい値を指定せずに、**-p** を組み合わせて使用する場合は、Tunable の現行値とリポート値が等しい場合のみ、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

-o、**-d** または **-D** を組み合わせて使用する場合は、現行値とリポート値の両方を変更します。つまり、現行値の更新だけでなく、**/etc/tunables/nextboot** ファイルの更新もオンにします。これらの組み合わせは **Reboot** および **Bosboot** タイプのパラメーターには使用できません。それらの現行値は変更できないためです。

新しい値を指定せずに、**-a** または **-o** を指定する場合は、そのパラメーターの現行値とリポート値が等しい場合のみ、値が表示されます。そうでない場合は、値に「NONE」が表示されます。

-r

-r フラグが、新しい値を指定せずにオプションの **-a** または **-o** と組み合わせて使用されると、パラメーターの現行値と次のリポート値が等しい場合にのみ、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** フラグは、フラグ **-o**、**-d**、または **-D** と組み合わせて使用された場合に、リポート値を変更します。例えば、**-r** フラグを使用した場合、**/etc/tunables/nextboot** ファイルを更新することができます。**Bosboot** タイプのどのパラメーターが変更されても、**bosboot** コマンドを実行するようユーザーにプロンプトが出されます。

新しい値を指定せずに、**-a** または **-o** オプションを指定する場合は、現行値のかわりに、Tunable のリポ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、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,min,max,unit,type,{dtunable }

ここで、

current = 現行値
 default = デフォルト値
 reboot = リポート値
 min = 最小値
 max = 最大値
 unit =
 チューナブル計測単位
 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 = list of dependent tunable parameters

項目	説明
-y	bosboot コマンドを実行する前の確認プロンプトを抑制します。

制限付きチューナブル・パラメーターが変更されると、制限付き使用タイプのチューナブルが変更されたことを示す警告メッセージが表示されます。**-r** または **-p** のオプションが指定されると、変更を確認するようプロンプトが出されます。さらに、システム・リブート時に、**/etc/tunables/nextboot** ファイルで表示された制限付きチューナブルは、そのデフォルト値とは異なる値に変更されており (**-r** オプションまたは **-p** オプションを指定するコマンド・ラインを使用)、これらの変更されたチューナブルのリストを識別するエラー・ログに入力されます。

チューナブルを変更する場合、単位を示すため、チューナブル値に K、M、G、T、P および E のように省略形で指定することがあります。省略形と、それに対応する値については、以下のリストを参照してください。

- K=2¹⁰
- M=2²⁰
- G=2³⁰
- T=2⁴⁰
- P=2⁵⁰
- E=2⁶⁰

このように、チューナブル値 1024 は 1 K と指定します。

Mount タイプのパラメーターを変更 (**-o**、**-d** または **-D**) すると、後のマウントに対してのみ変更が有効であることをユーザーに警告するメッセージが表示されます。

Connect タイプのパラメーターを変更 (**-o**、**-d** または **-D** フラグ) すると、**inetd** がリスタートとなり、後のソケット接続に対してのみ変更が有効であることをユーザーに警告するメッセージが表示されます。

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

Incremental タイプのパラメーターの現行値を、現行値より小さい新規の値を指定して、(**-o**、**-d** または **-D** を指定し、**-r** は指定せずに) 変更しようとする、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・タイプ

チューニング・コマンド (**no**、**nfso**、**vmo**、**ioo**、**raso**、および **schedo**) で操作されたチューナブル・パラメーターはすべて、以下のカテゴリーに分類されます。

項目	説明
Dynamic	パラメーターがいつでも変更可能な場合
Static	パラメーターが変更不可の場合
Reboot	リブート中のみパラメーターの変更が可能な場合
Bosboot	Bosboot の実行とマシンのリブートでのみパラメーターの変更が可能な場合
Mount	パラメーターの変更が、後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対してのみ有効な場合
Incremental	ブート時を除き、パラメーターの増加のみ可能な場合
Connect	パラメーターへの変更が、今後のソケット接続に対してのみ有効な場合
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

Bosboot タイプのパラメーターに関しては、変更が実行される時はいつでも、チューニング・コマンドが自動的にユーザーにプロンプトを出して **bosboot** コマンドを実行するかどうかを尋ねます。Connect タイプのパラメーターに関しては、チューニング・コマンドが **inetd** デーモンをリスタートさせます。

vmo が管理するパラメーターの現行のセットには、Dynamic および Bosboot 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

互換モード (**sys0** の **pre520tune** 属性で制御されるモード) で実行している場合は、Bosboot タイプ以外のパラメーターにはリブート値は、実際には意味がありません。このモードでは、それらの値はブート時に適用されないためです。詳しくは、以下を参照してください。パフォーマンス・マネージメント。

互換モードでは、ブート・シーケンスの間に呼び出されるスクリプト内にチューニング・コマンドへの呼び出しを埋め込んで、リブート値をチューニング・パラメーターへ設定することができます。**Reboot** タイプのパラメーターは、**-r** フラグを使用せずに設定できるため、既存のスクリプトは動作し続けます。

チューナブル・パラメーター

チューナブルに使用できるデフォルト値と値の範囲を表示するには、以下のように **-h** オプションを指定して **vmo** コマンドを実行します。

```
vmo -h <tunable_parameter_name>
```

チューナブル	説明
ame_cpus_per_pool	<p>目的</p> <p>圧縮メモリー・プール当たりの CPU の率を決定します。どの <i>ame_cpus_per_pool</i> CPU にも、少なくとも 1 個の圧縮メモリー・プールが作成されます。</p> <p>チューニング</p> <p>圧縮メモリー・プールでの競合を減らしたい場合は、低い率を使用します。この率は、圧縮メモリー・プール数の決定に使用される唯一の要因ではありません (メモリー量とレイアウトも考慮されます)。したがって、この率に対してなんらかの変更をしても、圧縮メモリー・プール数はまったく変更されない可能性があります。</p>
ame_maxfree_mem	<p>目的</p> <p>圧縮メモリー・プールのフリー・リスト内の空きメモリーの平均量の最大値を指定します。この値に達すると、VMM は圧縮プールを縮小します。</p> <p>チューニング</p> <p>圧縮メモリー・プール・サイズが著しく変化する傾向がある場合は、過度の縮小操作と拡張操作が発生する可能性があります。これは、ワークロードの作業セットのサイズが頻繁に変更される場合に発生することがあります。このチューナブルの値を大きくすると、VMM が圧縮メモリー・プールを縮小するしきい値が上がります。縮小操作と拡張操作の全体数を減らすことができます。</p>

チューナブル	説明
ame_min_ucpool_size	<p>目的</p> <p>非圧縮プールの最小サイズを定義します。</p> <p>チューニング</p> <p>圧縮メモリー・プールのサイズが大きくなりすぎると、メモリー内のスペースが非圧縮メモリーを格納するのに十分でなくなる場合があります。それによって圧縮メモリー・プールが過度に使用されるため、アプリケーションのパフォーマンスがスローダウンする可能性があります。圧縮メモリー・プールのサイズを制限し、使用可能な非圧縮ページ数を増加するには、この値を大きくしてください。</p>
ame_minfree_mem	<p>目的</p> <p>圧縮メモリー・プールのフリー・リスト内の空きメモリーの平均量の最小値を指定します。この値まで減少すると、VMM は圧縮プールを拡張します。</p> <p>チューニング</p> <p>圧縮メモリーが使用可能になるのを待ってプロセスが遅延している場合は、応答時間を改善するために <i>ame_minfree_mem</i> を増加します。この値には、<i>ame_maxfree_mem</i> より少なくとも 64 KB 小さい値を指定する必要があることに注意してください。</p>
ame_mpsize_support	<p>目的</p> <p>POWERS[®] プロセッサ・ベース・サーバーまたはそれ以降について、64 KB アクセラレーターをサポートする Active Memory Expansion (AME) 環境でサポートされるすべてのページ・サイズを使用可能にします。</p> <p>チューニング</p> <p>値 0 は、AME 環境で従来の動作を有効にします。この場合、有効になるのは、4 KB と 16 MB のページ・サイズのみです。値 1 の場合は、AME 環境でサポートされているすべてのページ・サイズが有効となります。このチューナブル・パラメーターは、64 KB アクセラレーターをサポートする、POWERS プロセッサ・ベース・サーバーまたはそれ以降でのみ変更可能です。</p>
ams_loan_policy	<p>目的</p> <p>このチューナブルは、共用メモリー・モードが使用可能になっているときに、ローニング動作を切り替えます。</p> <p>チューニング</p> <p>このチューナブルが 0 に設定されている場合、ローニングは使用不可です。1 に設定すると、ファイル・キャッシュのローニングが使用可能になります。2 に設定すると、あらゆるタイプのデータのローニングが使用可能になります。AMS プール内のメモリー量が少なくなっている場合、VMM はそれに応答してメモリーを解放し、ハイパーバイザーにそのメモリーをローニングします。</p>

チューナブル	説明
force_realias_lite	<p>目的</p> <p>0 に設定されている場合、mmap 領域を解消する時に、ソースの mmap されている セグメントのロックを回避する時期を判別するために、ヒューリスティックが使用されます。</p> <p>チューニング</p> <p>これは、<i>realias_percentage</i> によって制御される度合の指定 (計算時間がかかる可能性がある) をオフにする働きがあります。1 に設定されている場合は、<i>realias_percentage</i> の値に関係なく、可能な限りソース・セグメントのロックを回避します。</p>
kernel_heap_psize	<p>目的</p> <p>カーネル・ヒープに使用するデフォルトのページ・サイズを指定します。</p> <p>チューニング</p> <p>これは設定を推奨します。64 KB ページのサポートは、POWER5+ 以降のマシンによって提供され、<i>vmm_mpsize_support</i> が使用可能になっている場合に使用されます。POWER4 以降のマシンによって提供されている 16 MB ページは、ハイパフォーマンス環境のカーネル・ヒープにのみ使用します。値 0 は、優先デフォルト値 64 KB のページ・サイズがサポートされている場合にカーネルがこのページ・サイズを使用することを示しています。このページ・サイズがサポートされていない場合は、4 KB ページが使用されます。</p>
lgpg_regions	<p>目的</p> <p>SHM_LGPAGE フラグを指定した shmget() システム・コールを使用する実装のために予約するラージ・ページの数を指定します。</p> <p>チューニング</p> <p>このオプションに加えて、<i>lgpg_size</i> パラメーターも使用する必要があります。shmget() を呼び出す場合に、SHM_LGPAGE フラグを指定するようアプリケーションを変更する必要があります。これにより、多くの TLB ミスがあり、大量のメモリーがアクセスされているような場合、パフォーマンスが改善されます。</p> <p>このパラメーターは DLPAR 対応システムでは Dynamic ですが、bosboot コマンドの実行時に nextboot 値がブート・イメージに書き込まれるので、リポート時に最適な設定がリストアされます。</p>

チューナブル	説明
lgpg_size	<p>目的</p> <p>SHM_LGPGPAGE フラグを指定した shmget() システム・コールのインプリメンテーションに使用される、ハードウェアがサポートするラージ・ページのサイズ (バイト数) を指定します。</p> <p>チューニング</p> <p>POWER4 以降のシステムでサポートされます。このパラメーターは DLPAR 対応システムでは Dynamic ですが、bosboot コマンドの発行時に nextboot 値がブート・イメージに書き込まれるので、リポート時に最適な設定がリストアされます。このパラメーターに加え、lgpg_regions パラメーターをゼロ以外の値に設定する必要があります。shmget() サブルーチン呼び出し場合に、SHM_LGPGPAGE フラグを指定するようアプリケーションを変更する必要があります。これにより、多くの TLB ミスがあり、大量のメモリーがアクセスされているような場合、パフォーマンスが改善されます。</p>
low_ps_handling	<p>目的</p> <p>低ページング・スペース状態でのプロセス終了に関連するシステム動作を変更するためのアクションを指定します。</p> <p>チューニング</p> <p>値が 1 の場合、低ページング・スペース時に、プロセス終了を行うという現行の動作を示します。値が 2 の場合は、低ページング・スペース状態から回復するまでにその他のプロセスが見つからなかった場合に、SIGDANGER ハンドラーをもつプロセスが強制終了されるという新規の動作を示しています。</p>
maxfree	<p>目的</p> <p>フリー・リスト上のフレームの最大数を指定します。この数に達するとページ・スチールは停止されます。</p> <p>チューニング</p> <p>vmstat -n コマンドを使用してフリー・リスト・サイズの変更を監視します。vmstat -n コマンドの表示から、アプリケーションの要求によってフリー・リストのサイズが頻繁に minfree を下回っていることが示される場合は、maxfree の値を大きくして、フリー・リストを補充するためのコールの数を減らしてください。この値を高く設定しすぎると、ページ置換が長時間実行されることとなります。maxfree と minfree の差は、ほぼ maxpgahead 程度で、8 以上でなければなりません。</p>

チューナブル	説明
maxpin%	<p>目的</p> <p>ピンできる実メモリーの最大パーセントを指定します。</p> <p>チューニング</p> <p>空きメモリーが使用可能なのにメモリーをピンできない場合は変更します。この値を変更する場合は、必ず新規値では、カーネルが使用するために実メモリーの少なくとも 4 MB をピン解除したままにしておいてください。vmo コマンドは、maxpin% を対応する maxpin 絶対値に変換します。これがカーネルによって使用される値です。このパラメーターは、最大負荷のベンチマークなどの極端な状態でのみ変更してください。</p> <p>bosboot コマンドが発行されると、この動的パラメーターの nextboot 値がブート・イメージに書き込まれます。</p>
memory_frames	<p>目的</p> <p>有効なメモリー・フレームの数。</p> <p>チューニング</p> <p>該当なし</p>
memplace_data	<p>目的</p> <p>データのデフォルトのメモリー配置ポリシーを指定します。</p> <p>チューニング</p> <p>主実行可能モジュール (初期化されたデータ、BSS)、ヒープ、共用ライブラリー、および実行時にロードされたオブジェクト・モジュールのデータを参照します。データ配置は、ファースト・タッチ (値 1)、システム全体でラウンドロビン (値 2)、あるいはシステムによってメモリーの最適な配置が決定される、自動 (値 0) に設定できます。</p>
memplace_mapped_file	<p>目的</p> <p>(shmat() および mmap() などによって) プロセスのアドレス・スペースにマップされるファイルのデフォルトのメモリー配置ポリシーを指定します。</p> <p>チューニング</p> <p>メモリー・マップ・ファイル配置のポリシーのデフォルトは、ファースト・タッチ (値 1)、システム全体でラウンドロビン (値 2)、あるいは、システムによってメモリーに最適な配置が決定される、自動 (値 0) に設定できます。</p>

チューナブル	説明
memplace_shm_anonymous	<p>目的</p> <p>匿名共用メモリのデフォルトのメモリ配置ポリシーを指定します。</p> <p>チューニング</p> <p>匿名共用メモリは、shmget() または mmap() によって作成された作業記憶域を指します。これには作成プロセスかその子孫しかアクセスできません。このメモリは名前 (またはキー) に関連付けられていません。匿名共用メモリのデフォルト配置は、システム全体でファースト・タッチ (値 1) またはラウンドロビン (値 2)、あるいは、システムによってメモリに最適な配置が決定される、自動 (値 0) に設定できます。</p>
memplace_shm_named	<p>目的</p> <p>名前付き共用メモリのデフォルトのメモリ配置ポリシーを指定します。</p> <p>チューニング</p> <p>名前付き共用メモリは、shmget() または shm_open() によって作成された作業記憶域を指します。これは名前 (すなわちキー) と関連付けられているので、この名前 (キー) を使用して複数のプロセスがこのメモリに同時にアクセスできます。名前付き共用メモリのデフォルト配置は、システム全体でファースト・タッチ (値 1) またはラウンドロビン (値 2)、あるいは、システムによってメモリに最適な配置が決定される、自動 (値 0) に設定できます。</p>
memplace_stack	<p>目的</p> <p>プログラム・スタックのデフォルトのメモリ配置ポリシーを指定します。</p> <p>チューニング</p> <p>スタック配置は、システム全体でファースト・タッチ (値 1) またはラウンドロビン (値 2)、あるいは、システムによってメモリに最適な配置が決定される、自動 (値 0) に設定できます。</p>
memplace_text	<p>目的</p> <p>アプリケーション・テキストのデフォルトのメモリ配置ポリシーを指定します。</p> <p>チューニング</p> <p>これは主実行可能モジュールのテキストにのみ適用され、その従属には適用されません。テキスト配置は、システム全体でファースト・タッチ (値 1) またはラウンドロビン (値 2)、あるいは、システムによってメモリに最適な配置が決定される、自動 (値 0) に設定できます。</p>

チューナブル	説明
memplace_unmapped_file	<p>目的</p> <p>read0/write0 など、マップ解除されたファイル・アクセスのデフォルトのメモリー配置ポリシーを指定します。</p> <p>チューニング</p> <p>マップ解除されたファイル・アクセスのデフォルト配置は、システム全体でファースト・タッチ (値 1) またはラウンドロビン (値 2)、あるいは、システムによってメモリーに最適な配置が決定される、自動 (値 0) に設定できます。</p>
minfree	<p>目的</p> <p>フリー・リスト上のフレーム数の最小値を指定します。この値まで減少すると、VMM はフリー・リストを補充するためにページをスチールし始めます。</p> <p>チューニング</p> <p>ページ置換は、空きフレームの数が <i>minfree</i> に達した時に起こります。ページ・スチールによってプロセスが遅延している場合は、応答時間を改善するために <i>minfree</i> を増やしてください。<i>maxfree</i> と <i>minfree</i> の差は、ほぼ <i>maxpagehead</i> 程度で、8 以上でなければなりません。</p>
minperm%	<p>目的</p> <p>これを下回ると、ページング率に関係なく、ページ・スチール機能がファイルまたは計算ページをスチールするポイントを指定します。</p> <p>チューニング</p> <p>メモリー内のファイル・ページ数が多いために作業記憶域ページの置換が発生している場合は、このパラメーターを減らすことができます。一方、繰り返し読み取られているファイルがいくつかあり、入出力率が始動時から時間とともに減少しない場合は、<i>minperm</i> が小さすぎる可能性があります。</p>
nokilluid	<p>目的</p> <p>この値より下のユーザー ID は、低ページ・スペース状態が原因で強制終了されることはありません。</p> <p>チューニング</p> <p>値 0 はオフを表します。システムのページング・スペースがなくなって、システム管理プロセスが強制終了されているときに有用です。低ページ・スペースが原因で特定のユーザー ID プロセスが強制終了されることのないようにこのチューナブルを 1 に設定するか、使用可能なページング・スペースを十分に確保してください。</p>

チューナブル	説明
npskill	<p>目的</p> <p>オペレーティング・システムがプロセスの強制終了を開始する空きページング・スペース・ページの数を指定します。</p> <p>チューニング</p> <p>デフォルト値は、64 と (ページング・スペースのページ数) / 128 の大きいほうです。<i>npskill</i> の値は、ゼロより大きく、かつシステム上のページング・スペース・ページの合計数より小さくしなければなりません。</p>
npswarn	<p>目的</p> <p>オペレーティング・システムが SIGDANGER シグナルをプロセスに送信し始める空きページング・スペース・ページの数を指定します。</p> <p>チューニング</p> <p>デフォルト値は、512 と ($4 * npskill$) の大きいほうです。<i>npswarn</i> の値は、ゼロより大きく、かつシステム上のページング・スペース・ページの合計数より小さくしなければなりません。低ページング・スペースが原因でプロセスが強制終了される場合は、この値を大きくしてください。</p>
numpsblks	<p>目的</p> <p>ページング・スペース・ブロックの合計数。</p> <p>チューニング</p> <p>該当なし</p>
pinnable_frames	<p>目的</p> <p>ピンに使用できるページ数。</p> <p>チューニング</p> <p>該当なし</p>

チューナブル	説明
relalias_percentage	<p>目的</p> <p><i>force_relalias_lite</i> が 0 に設定されている場合、これはソースの mmap されたセグメントのロックを回避するかどうかを決めるためにヒューリスティックで使用される要因を指定します。</p> <p>チューニング</p> <p>これは、mmap された領域が破壊されるときに使用されるスケールビリティ・ステートメントで、これによりロックが回避されるのでシステムのスループットが向上する可能性があります。場合によっては使用される計算時間が長くなります。マップ解除されるページ数が、この値を 100 で除算し、mmap されたソース・セグメント内のメモリー内のページ総数を乗算した数値よりも小さい場合、ソースのロックは回避されます。</p> <p><i>relalias_percentage</i> の値が 0 で、<i>force_relalias_lite</i> も 0 に設定されている場合、ソース・セグメントのロックは必ず発生します。<i>relalias_percentage</i> の有効値はワークロードによって異なりますが、推奨値は 200 です。</p>
scrub	<p>目的</p> <p>「据え置きページ・スペース割り振りポリシー」ページ用にメモリー内のページからページング・スペース・ディスク・ブロックを解放することを使用可能または使用不可にします。</p> <p>チューニング</p> <p>値 0 は消し込みを完全に使用不可にします。値 1 が指定されている場合は、システム・フリー・ページング・スペース・ブロックの数が npsscrubmin 未満であれば、メモリー内のページング・スペース・ディスク・ブロックの消し込みが可能になり、npsscrubmax の値を上回るまで継続されます。</p>
v_pinshm	<p>目的</p> <p>1 に設定されると、共用メモリー・セグメントのピンが可能になります。</p> <p>チューニング</p> <p>値 0 はオフを表します。共用メモリー・セグメントから AIO バッファをピンまたはピン解除する際のオーバーヘッドが高い場合は、この値を変更してください。shmget() 呼び出しの実行時にアプリケーションが SHM_PIN フラグも設定する場合や、共用メモリー・セグメントからの非同期入出力を実行している場合にのみ有用です。</p>

チューナブル	説明
vmm_default_pspa	<p>目的</p> <p>このチューナブルは、ページ・サイズ・プロモーションのデフォルトのアグレッシブネスを制御します。この値は、オペレーティング・システムによってページ・プロモーションしきい値の逆として扱われる抽象的なアグレッシブネスの重み付けです。</p> <p>チューニング</p> <p><i>vmm_default_pspa</i> 設定に値 0 を設定することは、100% のページ・プロモーションしきい値と同等です。すなわち、プロモートされるためには、メモリー範囲は 100% の実メモリーを占有している必要があります。<i>vmm_default_pspa</i> 設定に値 100 を設定することは、0 % のページ・プロモーションしきい値と同等です。すなわち、メモリー範囲はその範囲内のメモリーが初めて参照されると即時プロモートされます。<i>vmm_default_pspa</i> 設定に値 -1 を設定することは、ページ・プロモーションしきい値 -1 と同等です。すなわち、メモリー範囲のページ・プロモーションは決して行われません。ページ・サイズ・プロモーションしきい値が考慮されるのは、セグメント作成時のみです。したがって、<i>vmm_default_pspa</i> の変更は、チューナブルの調整後に作成されたセグメントのページ・サイズ・プロモーションしきい値にのみ影響します。</p>
wlm_memlimit_nonpg	<p>目的</p> <p>非ページング可能ページ・サイズ (16M、16G) を WLM <i>realmem</i> カウントと <i>virtmem</i> カウントに含めるかどうかを選択します。1 を選択すると、非ページング可能ページ・サイズは <i>realmem</i> 限界カウントと <i>virtmem</i> 限界カウントに含められません。0 を選択すると、ページング可能ページ・サイズ (4K、64K) のみが、<i>realmem</i> カウントと <i>virtmem</i> カウントに含められます。この値は、WLM Memory アカウンティングがオフの場合にのみ変更できます。さもないと変更は失敗します。</p> <p>チューニング</p> <p>このチューナブルが 0 に設定されている場合、WLM 仮想メモリー制限と実メモリー制限は、WLM クラスによって消費されるページング可能ページにのみ適用されます。ページング可能ページを頻繁に使用するとシステム上でページングが発生するため、値 0 は、非ページング可能ページが使用されているときに WLM クラスが行うページングの量について、より細かい制御を提供します。このチューナブルは、非ページング可能ページによって構成されているシステム上で WLM 実メモリー制限または仮想メモリー制限が使用されている場合にのみ調整する必要があります</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. **vmo** コマンドで管理されるチューナブル・パラメーターすべての、現行値、リブート値、範囲、単位、タイプ、および依存関係をリストするには、下記のように入力します。

```
vmo -L
```

2. POWER4 システムの 16 MB ラージ・ページを作動させて予約するには、下記のように入力します。

```
vmo -o lgpg_regions=10 -o lgpg_size=16777216
```

このコマンドはユーザーに **bosboot** を提案し、変更を有効にするにはリブートが必要であることを警告します。

3. **nokilluid** に関するヘルプを表示するには、下記のように入力します。

```
vmo -h nokilluid
```

4. 次のリブート後に **v_pinshm** をオンにするには、下記のように入力します。

```
vmo -r -o v_pinshm=1
```

5. すべての **vmo** チューナブル・パラメーターを永続的にデフォルトにリセットするには、下記のように入力します。

```
vmo -p -D
```

6. すべての仮想メモリー・マネージャー (VMM) のチューニング・パラメーター用のリブート値をリストするには、下記のように入力します。

```
vmo -r -a
```

7. **vmo** コマンドで管理される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、範囲、単位、タイプ、および依存関係を (スプレッドシート形式で) リストするには、下記のように入力します。

```
vmo -x
```

関連情報:

ioo コマンド

schedo コマンド

no コマンド

nfso コマンド

raso コマンド

vmstat コマンド

目的

仮想メモリーの統計情報を報告します。

構文

```
vmstat [ -f ] [ -i ] [ -s ] [ -I[-W] ] [ -t ] [ -v ] [ -h ] [ -w ] [ -l ] [ -c ] [ -@ wparname | ALL ] [ -p | -P ] [ physicalvolume | ALL ] ALL [ -S power ] [ physicalvolume ] [ interval ] [ count ]
```

注: *wparname* パラメーターと **-i** フラグは、workload partitions内で併用しないでください。

説明

vmstat コマンドは、カーネル・スレッド、仮想メモリー、ディスク、ハイパーバイザーのページ、トラップおよびプロセッサのアクティビティーに関する統計情報を報告します。 **vmstat** コマンドによって生成される報告は、システム・アクティビティーの負荷を調整するのに使用することができます。これらのシステム全体の統計情報 (すべてのプロセッサ間) は、パーセントで表される平均値または合計として計算されます。統計情報は原子的に読み取られないため、**vmstat** コマンドは不整合な統計情報を戻すことがあります。

vmstat コマンドをフラグなしで実行すると、システム始動後の仮想メモリー・アクティビティーの要約が報告されます。 *-f* フラグを指定すると、**vmstat** コマンドはシステム始動後の *fork* の数を報告します。 *physicalvolume* パラメーターには、物理ボリューム名を指定します。

interval パラメーターは、各レポートの間隔を秒で指定します。 *interval* パラメーターを指定しない場合、**vmstat** コマンドはシステム起動時以降の統計を含む単一のレポートを生成して終了します。 *count* パラメーターは、*interval* パラメーターと一緒に指定する必要があります。 *count* パラメーターを指定すると、その値によって、生成されるレポートの数と間隔 (秒数) が決まります。 *count* パラメーターなしで *interval* パラメーターを指定すると、レポートは連続して生成されます。 *count* パラメーターに、ゼロの値を指定しないでください。

カーネルは、カーネル・スレッド、ページング、および、割り込みのアクティビティーに関する統計情報を保守します。**vmstat** コマンドは、*perfstat* カーネル・エクステンションを使用して、これにアクセスします。ディスク入出力統計情報は、デバイス・ドライバーが保守します。ディスクの平均転送率は、活動時間と転送数の情報から求められます。活動時間 (パーセント) は、報告中にドライブがビジーであった時間から求められます。

vmstat コマンドは、マイクロ・パーティショニング環境において、消費された物理プロセッサの数 (*pc*)、および消費されたライセンスのパーセンテージ (*ec*) を報告するようになりました。マイクロ・パーティショニング環境に関するメトリックが表示されます。

vmstat コマンドで生成されるレポートには、システム構成の行と列の見出しが含まれています。 *-@* フラグを指定すると、レポートの構成は、システム構成および WPAR 構成になります。システム構成の行には、次の値があります。

lcpu 論理プロセッサの数を表示します。

mem メモリー容量を表示します。

tmem LPAR の true メモリー・サイズ。

注: このフラグは、*-c* オプションが指定されていて、Active Memory™ Expansion が使用可能な場合のみ有効です。

ent ライセンス済みキャパシティーを示します。区画が共用プロセッサで実行されている場合のみ表示されます。

drives

ディスク数を示します。物理ボリューム名のモニター中のみ表示されます。

WPAR

活動中の *workload partitions* の数を表示します。 *-@* フラグが指定された場合のみ表示されます。

memlim

workload partition のメモリー・リソースの制限を示します。この制限は、メガバイト (MB) 単位です。この情報は、メモリー・リソースの制限を強制した WPAR の場合にのみ表示されます。

cpulim

処理装置内の workload partition のプロセッサ・リソースの制限を示します。この情報は、プロセッサ・リソースの制限を強制した WPAR の場合にのみ表示されます。

rset WPAR と関連付けられている **rset** レジストリーのタイプを示します。このタイプには、**regular** (通常) または **exclusive** (排他) があります。この情報は、**rset** レジストリーと関連付けられている WPAR の場合にのみ表示されます。

mmode

メモリー・モードを示します。このメトリックは、アクティブ・メモリー共有が使用可能なシステムでは自動的に表示されます。このメトリックは、**-c** オプションが使用されているときにも表示されます。

mpsz メモリー・プールのサイズ (ギガバイト (GB) 単位)。このメトリックは、共有メモリー・モードでのみ表示されます。

以下は、列見出しとその説明です。

WPAR: workload partitions に関する情報。 **-@** フラグが指定された場合のみ表示されます。

WPAR

Workload partition 名。

注:

1. **system** WPAR 名は、システム全体の統計情報を示します。 **global** WPAR 名は、Global に属する統計情報のみを示します。
2. **vmstat** コマンドが **-@ ALL** オプションを使用して開始され、WPAR 固有の情報がメトリックとして使用できない場合、値に代わりダッシュ (-) が表示されます。
3. **vmstat** コマンドが **-@ wparname** で開始されるか、WPAR 内で開始された場合で、WPAR 情報がメトリックとして使用できない場合、そのメトリックはアットマーク (@) でマークされ、そのメトリックに対してはシステム全体の値が表示されます。
4. メトリックがサポートされない場合、値に代わりダッシュ (-) が表示されます。

kthr: カーネル・スレッドの状態に関する情報。

r サンプル間隔中における実行可能カーネル・スレッドの平均数。「実行可能スレッド」とは、作動可能であっても実行のためにまだ待機しているスレッドと、既に実行中のスレッドを含みます。

b サンプル間隔中に Virtual Memory Manager (VMM) 待機キュー (リソースを待機中、入出力を待機中) に入れられたカーネル・スレッドの平均数。

Memory: 仮想メモリーと実メモリーの使用状況に関する情報。仮想ページは、これまでにアクセスされたことがあれば、アクティブであると見なされます。1 ページは 4096 バイトです。

avm アクティブな仮想ページ。

fre フリー・リストのサイズ。

注: 実メモリーの大部分は、ファイル・システム・データのキャッシュとして使用されます。フリー・リストのサイズが小さいままであることは、異常ではありません。

Page: ページ・フォールトとページング・アクティビティーに関する情報。この情報は、指定した間隔で平均をとり、1 秒当たりの単位数として表示されます。

- re** ページャーの入出力リスト。
- pi** ページング・スペースからページ・インされたページ数。
- po** ページング・スペースにページ・アウトされたページ数。
- fr** 空きのページ数。
- sr** ページ置換アルゴリズムによってスキャンされたページ数。
- cy** ページ置換アルゴリズムによるクロック・サイクル数。

Faults: サンプルング間隔中 1 秒当たりのトラップおよび割り込み率の平均値。

- in** デバイス割り込みの回数。
- sy** システム・コール。
- cs** カーネル・スレッドのコンテキスト・スイッチ数。

CPU: プロセッサ時間の使用率の内訳。

- us** ユーザー時間。
上限なし区画の現行物理プロセッサ消費量がライセンス済みキャパシティーを超える場合、そのパーセンテージは、消費される物理プロセッサ (pc) の数に関連するようになります。
- sy** システム時刻。
上限なし区画の現行物理プロセッサ消費量がライセンス済みキャパシティーを超える場合、そのパーセンテージは、消費される物理プロセッサ (pc) の数に関連するようになります。
- id** プロセッサのアイドル時間。
上限なし区画の現行物理プロセッサ消費量がライセンス済みキャパシティーを超える場合、そのパーセンテージは、消費される物理プロセッサ (pc) の数に関連するようになります。
- wa** システムに未解決のディスク/NFS I/O 要求があったプロセッサのアイドル時間。
上限なし区画の現行物理プロセッサ消費量がライセンス済みキャパシティーを超える場合、そのパーセンテージは、消費される物理プロセッサ (pc) の数に関連するようになります。
- pc** 使用された物理プロセッサの数。区画が共用プロセッサで実行されている場合のみ表示されます。
- ec** 消費されるライセンス済みキャパシティーのパーセンテージ。区画が共用プロセッサで実行されている場合のみ表示されます。データを計算する際の時間基準がさまざまであるために、ライセンス済みキャパシティーのパーセンテージが 100% を超えることがあります。このような超過の現象は、サンプルングの時間間隔が短い場合にのみ顕著に現れます。
- rc** 使用されるプロセッサ・リソースのパーセンテージ。この情報は、プロセッサ・リソースの制限を強制した WPAR の場合にのみ表示されます。

Disk: サンプル・インターバルにおける指定した物理ボリュームへの転送数 (毎秒) を示します。
physicalvolume パラメーターを使って、1 つから 4 つまでの名前を指定することができます。転送統計情報は、指定した各ドライブについて、指定した順番で報告されます。この値は、物理デバイスに対する要求を表します。読み取られた、または書き込まれたデータの数はありません。複数の論理要求を 1 つの物理要求にまとめることができます。*physicalvolume* パラメーターが使用される場合、コマンド実行の始めて物理ボリューム名が印刷されます。

-I フラグを指定すると、I/O 指向のビューでは、下記の列の変更が表示されます。

kthr 列 **r** および **b** に加えて、列 **p** も表示されます。

p ロウ・デバイスからの入出力メッセージを待機中の平均スレッド数。ロウ・デバイスは、システムに直接接続されたデバイスです。

-I フラグと共に **-W** フラグが指定されていると、フラグ **r**、**b** および **p** と共に、追加の桁 **w** も表示されます。

w ファイルシステムの直接入出力イベントが発生するのを待機している、1 秒当たりのスレッドの数。これらのイベントには、以下のタイプのものがあります。

- 非同期入出力 (AIO)
- バッファ・キャッシュ・サブシステム
- 並行入出力 (CIO)
- ファイルシステム直接入出力
- NFS サブシステム
- スレッドが、仮想メモリー・マネージャー (VMM) 待ちリストからのアクションを待機しています。

page 新規の列 **fi** および **fo** は、**re** および **cy** 列の代わりに表示されます。

fi 秒ごとのファイル・ページ・イン。

fo 秒ごとのファイル・ページ・アウト。

-c フラグを指定すると、Active Memory Expansion ビューでは、下記の列の変更が表示されます。

memory

列 **csz**、**cfr**、および **dxm** は、列 **avm** および **fre** の隣に表示されます。

csz 現在の圧縮プール・サイズ (4K ページ単位)。

cfr 圧縮プール内の使用可能なフリー・ページ数 (4K ページ単位)。

dxm 拡張メモリー内の不足サイズ (4K ページ単位)。

page 新規の列 **ci** および **co** は、列 **re** および **cy** の代わりに表示されます。

ci 圧縮プールからのページ・インの数 (1 秒当たり)。

co 圧縮プールへのページ・アウトの数 (1 秒当たり)。

vmstat コマンドの実行中に、出力に影響を及ぼす変更がシステム構成内にある場合、**vmstat** は構成変更についての警告メッセージを表示します。コマンドは、更新済みのシステム構成情報およびヘッダーを表示した後で、出力を継続します。

-l フラグを指定すると、以下の列を含む追加の「ラージ・ページ」セクションが表示されます。

alp 現在使用中のラージ・ページの数を示します。

flp ラージ・ページのフリー・リストにあるラージ・ページの数を示します。

-p オプションが指定されている場合は、指定されたページ・サイズに固有の VMM 統計情報の行が補足的に表示されます。 **-I** および **-t** オプションとともに、**-p** オプションは指定されたページ・サイズについての行を補足的に生成します。この行には、指定されたページ・サイズと関係のある次の VMM 統計情報が含まれます。

- **avm**
- **fre**

- re
- fi
- fo
- pi
- po
- ci
- co
- fr
- sr
- cy

注:

1. re、fi、fo、および cy オプションの表示は、-I オプションの影響を受けます。
2. re、ci、co、および cy オプションの表示は、-c オプションの影響を受けます。
3. リソース制御がない場合、avm および fre オプションはシステム規模のもので、したがって、-@ オプションを設定すると、avm と fre の両オプションがアットマーク (@) でマーク付けられます。

これらの VMM 統計情報は psz 列の後、siz 列の前に表示されます。この 2 つの列について次に説明します。

psz ページ・サイズ (例えば、4 KB、64 KB)。

siz システムに存在する指定されたページ・サイズのフレームの数。

-s オプションと共に -p オプションを指定すると、別の出力スタンザが作成され、ここには指定されたページ・サイズと関係のある統計情報のみが収容されます。この追加のスタンザはページ・サイズ・ヘッダーに続いて出力されます。

-P オプションは指定されたページ・サイズに対して次のような報告書を作成します。

pgsz ページ・サイズ (例えば、4 KB、64 KB) を示します。

Memory

指定されたページ・サイズに対するメモリー統計情報を示します。

siz システムに存在する指定されたページ・サイズのフレームの数。

avm 指定されたページ・サイズに対して利用可能な活動中の仮想ページ。

fre 指定されたページ・サイズに対するフリー・リストのサイズ。

Page 指定されたページ・サイズに関係のあるページ・フォールトとページ送りの活動状況を示します。ページに関連する列 re、pi、po、fr、sr、cy、fi、fo、ci、および co もこの報告書に適用できます。

フラグ

注: コマンド・ラインに -f (または -s) フラグを入力すると、システムは -f (または -s) フラグのみを受け入れ、他のフラグは無視します。-f フラグと -s フラグを両方指定すると、システムは最初のフラグのみを受け入れ、2 番目のフラグは無視します。

項目	説明
-@ <i>wparname</i>	<p>workload partitionの仮想メモリーの活動状況をレポートします。</p> <ul style="list-style-type: none"> • -@ ALL オプションは、レポートがシステム中のすべてのworkload partitionsに加え、システムとグローバル環境に関係するものであることを示します。 注: システム全体の統計情報の値は、WPAR セクションに照らしてダッシュ (-) でマークされます。 • -@<i>wparname</i> フラグは、アクティビティーがそのworkload partitionのみに対するものであることを示します。workload partition では、-@ フラグを指定すると、システム全体の統計情報および workload partition 統計情報が表示されます。システム全体の統計情報は、アットマーク (@) でマークされます。 注: -@ フラグを -i フラグと組み合わせて使用しないでください。
-c	<p>メモリー圧縮統計情報で、出力の新規の列を表示します。それらは、ヘッダー Memory の下の列 csz、cfr および dxm、およびヘッダー・ページの下の列 ci および co (ヘッダー・ページの列 re および cy に代わるもの) です。 注: このオプションは、Active Memory Expansion が使用可能な場合のみ有効です。</p>
-f	<p>システム始動後の fork の数を報告します。</p>
-i	<p>システム始動後に各デバイスが行った割り込みの数を表示します。 注: -I、-t、-w、および -l フラグは、-i フラグとともに指定されると無視されます。</p>
-I	<p>I/O 指向のビューで、出力の新規の列を表示します。それらは、ヘッダー kthr の下の p、ヘッダー・ページの下 fi および fo (ヘッダー・ページの列 re および cy に代わるもの) です。</p>
-l	<p>alp および flp 列を含む追加の「ラージ・ページ」セクションを表示します。</p>
-p <i>pagesize</i>	<p>指定されたページ・サイズに対する VMM 統計情報を、通常の vmstat 出力結果に追加します。</p>
-P <i>pagesize</i>	<p>指定されたページ・サイズに関係のある VMM 統計情報のみを表示します。</p>
-s	<p>合計構造体の内容を標準出力に書き出します。この合計構造体には、システム初期化後に発生したページング・イベントの絶対数が入っています。-s フラグは、-v フラグと一緒にないと使用することができません。以下に、これらのイベントを説明します。</p> <p>address translation faults</p> <p>アドレス変換ページ・フォールトが発生した回数です。ページ・フォールトを回復するために入出力が要求されていたかどうかは分かりません。この値には、ストレージ保護ページ・フォールト (ロックの失敗) は含まれません。</p>

説明

backtracks

以前のページ・フォールトの回復中に発生したページ・フォールトの回数です。まず最初に新しいページ・フォールトを回復する必要があるため、次に、初期ページ・フォールトを *backtracked* 可能です。

CPU context switches

プロセッサ・コンテキスト・スイッチ (新しいプロセスのディスパッチ) の回数です。

decrementer interrupts

decrementer 割り込みの回数です。

device interrupts

ハードウェア割り込みの回数です。

executable-filled page faults

命令のページ・フォールトの数です。

extend XPT waits

アクセス中のセグメントのコミットのために、VMM によってプロセスが待機状態になった回数です。

free frame waits

プロセスがページ・フレームを要求する回数です。フリー・リストが空になり、そしてフリー・リストが補充される間、プロセスが待機させられます。

iodones VMM の各入出力要求が完了した回数です。

mpc send interrupts

MPC 送信割り込みの回数です。

mpc receive interrupts

MPC 受信割り込みの回数です。

page ins

仮想メモリー・マネージャーがページを読み取った回数です。この値は、ページ・スペースおよびファイル・スペースのページ・インの回数を表します。この値とページ・アウト統計情報の値を合わせると、仮想メモリー・マネージャーが開始した実入出力の総数が求められます。

page outs

仮想メモリー・マネージャーがページを書き込んだ回数です。この値は、ページ・スペースおよびファイル・スペースのページ・アウトの回数を表します。この値とページ・イン統計情報の値を合わせると、仮想メモリー・マネージャーが開始した実入出力の総数が求められます。

paging space page ins

VMM が行ったページング・スペースだけのページ・インの回数です。

paging space page outs

VMM が行ったページング・スペースだけのページ・アウトの回数です。

pages examined by the clock

VMM は、クロック・アルゴリズムを使って、疑似 LRU ページ置き換えスキームを実現します。ページは、クロックに検査されることで古くなります。この値は、ページがクロックに検査された回数です。

pages freed by the clock

実メモリーから解放するために、クロック・アルゴリズムがページを選択した回数です。

pending I/O waits

ページ・イン入出力が完了するまでに、VMM によってプロセスが待機状態になった回数です。

説明

phantom interrupts

phantom 割り込みの回数です。

revolutions of the clock hand

VMM クロック回転数 (つまり、メモリーのスキャン数) です。

start I/Os

VMM が開始した読み取りまたは書き込み入出力要求の回数です。

syscalls システム・コールの回数です。

total reclaims

新しい入出力要求を実行しなくても回復できるアドレス変換障害の数です。この状態は、以前 VMM で要求されていたページの、入出力が完了していなかったときに発生します。また、VMM の先読み取りアルゴリズムであらかじめフェッチされていたページが不在の発生したセグメントには見えなかったときや、フリー・リストにあったページが再使用されていなかったときにも発生します。

traps このオペレーティング・システムではサポートしていません。

zero-filled page faults

作業用ストレージで発生し、フレームの割り当てとゼロ埋め込みで回復できるページ・フォールの数です。

-c フラグを -s フラグと共に指定すると、以下の追加メトリックが表示されます。

compressed pool page ins

システム・ブート以来の圧縮プールからのページ・インの数。

compressed pool page outs

システム・ブート以来の圧縮プールへのページ・アウトの数。

-s

-s オプションを -p *pagesize* オプションとともに使用すると、指定されたページ・サイズに対する合計情報をシステム全体の合計情報に追加できます。この追加のスタンザはページ・サイズ・ヘッダーに続いて出力されます (例えば、4K ページ)。次の統計情報はページ・サイズには関係ないため、このページ・サイズ・ベースのスタンザには出力されません。

- プロセッサ・コンテキスト切り換えの回数
- デバイス割り込みの回数
- ソフトウェア割り込みの回数
- decremter 割り込みの回数
- MPC-sent 割り込みの回数
- MPC-received 割り込みの回数
- phantom 割り込みの回数
- トラップの回数
- システム・コールの回数

注:

1. -@ ALL オプションとともに -s フラグを使用すると、システム全体の統計情報はworkload partitionセクションで繰り返されます。
2. *wparname* オプションとともに -s フラグを使用すると、すべてのメトリックが報告され、システム全体の統計情報はアットマーク (@) でマークされます。
3. -s フラグを -l フラグと共に使用すると、**vmstat** コマンドは次のメトリックを表示します。

large-page hi water count

ラージ・ページ使用中カウンットの最大値を指定します。

項目
-S *power*

説明
プロセッサの統計情報にべき乗の値 (10^{power}) を掛けます。べき乗のデフォルト値は 0 です。

以下の統計情報は、スケーリングされます。

- **us**
- **sy**
- **id**
- **wa**
- **pc**
- **ec**

注:

1. -S フラグは、-f、-s、-i、-v、または -p フラグと一緒に使用しないでください。
2. -S フラグが指定されると、**us**、**sy**、**id**、および **wa** の統計情報が変わります。デフォルトで、**us**、**sy**、**id**、および **wa** の統計情報は、WPAR のプロセッサ使用量と比較して表示されます。-S フラグとゼロでないべき乗の値が指定されると、これらの統計情報はシステム全体のプロセッサ使用量と比較して表示されます。
3. -S フラグのべき乗の値は、0 と 3 の間のみです。

-t

vmstat の出力の各行の隣に、タイム・スタンプを表示します。このタイム・スタンプは、HH:MM:SS フォーマットで表示されます。

注: -f、-s、または -i フラグが指定された場合、タイム・スタンプは表示されません。

項目
-v

説明

仮想メモリー・マネージャーが保守するさまざまな統計情報を標準出力へ書き込みます。 -v フラグは、 -s および -h フラグ と一緒に使用される場合のみ使用可能です。

-v フラグを指定すると、以下の統計情報が表示されます。

compressed percentage

圧縮されたページに使用されるメモリーのパーセンテージ。

client filesystem I/Os blocked with no fsbuf

fsbuf が使用可能でないためにブロックされた、クライアント・ファイルシステム入出力要求の数。 NFS (ネットワーク・ファイルシステム) および VxFS (Veritas) がクライアント・ファイルシステムです。 fsbuf は、ファイルシステム・レイヤーで入出力要求を保留するために使用される、ピンされたメモリー・バッファーです。

client pages

クライアント・ページの数。

compressed pages

圧縮されたメモリー・ページの数。

external pager filesystem I/Os blocked with no fsbuf

fsbuf が使用可能でないためにブロックされた、外部ペーজャーのクライアント・ファイルシステム入出力要求の数。 JFS2 が外部ペーজャーのクライアント・ファイルシステムです。 fsbuf は、ファイルシステム・レイヤーで入出力要求を保留するために使用される、ピンされたメモリー・バッファーです。

file pages

現在ファイル・キャッシュが使用している 4 KB ページの数。

free pages

フリーの 4 KB ページの数。

filesystem I/Os blocked with no fsbuf

fsbuf が使用可能でないためにブロックされた、ファイルシステム入出力要求の数。 fsbuf は、ファイルシステム・レイヤーで入出力要求を保留するために使用される、ピンされたメモリー・バッファーです。

lrutable pages

置換されると考えられる 4 KB ページの数。この数には、VMM 内部ページに使用されるページや、カーネル・テキストのピンされた部分に使用されるページは含まれません。

maxclient percentage

クライアント・ページが使用できるメモリーの最大パーセンテージを指定するチューニング・パラメーター (vmo を使用して管理される)。

maxperm percentage

実メモリーのパーセンテージで指定するチューニング・パラメーター (vmo を使用して管理される)。

maxpin percentage

ピンできる実メモリーのパーセンテージを指定するチューニング・パラメーター (vmo を使用して管理される)。

memory pages

4 KB ページの数で表した実メモリーのサイズ。

memory pools

メモリー・プール数を指定するチューニング・パラメーター (vmo を使用して管理される)。

minperm percentage

実メモリーのパーセンテージで指定するチューニング・パラメーター (vmo を使用して管理される)。

numclient percentage

クライアント・ページ数が占めるメモリーのパーセンテージ。

項目
-v

説明
(-v で表示される統計情報の続き):

numperm percentage

現在ファイル・キャッシュが使用しているメモリーのパーセンテージ。

paging space I/Os blocked with no psbuf

psbuf スペースが使用可能でないためにブロックされた、ページング・スペース入出力要求の数。 psbuf スペースは、仮想メモリー・マネージャー・レイヤーで入出力要求を保留するために使用される、ピンされたメモリー・バッファです。

pending disk I/Os blocked with no pbuf

pbuf が使用可能でないためにブロックされた、保留中のディスク入出力要求の数。 pbuf は、論理ボリューム・マネージャー・レイヤーで入出力要求を保留するために使用される、ピンされたメモリー・バッファです。

pinned pages

ピンされた 4 KB ページの数。

注: カーネル・ロック機能 (vmm_klock_mode パラメーター) が有効になっていると、ピンされたページにカーネル・ロック (**klocked**) されたページが含まれます。カーネル・ロック機能について詳しくは、コマンド `vmo -h vmm_klock_mode` を実行してください。

remote pageouts scheduled

クライアント・ファイルシステムにスケジュールされたページ・アウトの数。

-h フラグを -v フラグと共に指定すると、以下の追加メトリックが表示されます。

Time resolving virtualized partition memory page faults

メモリー・ページ・フォールトが解決されるまで待機するために、仮想区画がブロックされている合計時間。時間はミリ秒の細分度で、計測の単位は秒です。

Virtualized partition memory page faults

仮想化区画に記録された仮想区画メモリー・ページ・フォールトの総数。

Number of 4 KB page frames loaned

ハイパーバイザーに貸し出された区画のメモリーの 4 KB ページの数。

Percentage of partition memory loaned

ハイパーバイザーに貸し出された区画のメモリーのパーセンテージ。

注:

1. **-@ ALL** オプションとともに **-v** フラグを使用すると、システム全体の統計情報は `workload partition` セクションで繰り返されません。
2. `wparname` オプションとともに **-s** フラグを使用すると、すべてのメトリックが報告され、システム全体の統計情報はアットマーク (@) でマークされます。

-c フラグを -v フラグと共に指定すると、以下の追加メトリックが表示されます。

Compressed Pool Size

圧縮プールのサイズ (4K ページ単位)。

percentage of true memory that is used for compressed pool

圧縮プールに使用される不拡張メモリーのパーセンテージ。

free pages in compressed pool (4K pages)

圧縮プール内のフリー・ページの数 (4K ページ単位)。

target memory expansion factor

LPAR 用に構成済みのターゲット・メモリー拡張係数。

achieved memory expansion factor

現在の獲得メモリー拡張係数

項目	説明
-h	ハイパーバイザーのページ情報を含む hypv-page セクションを表示します。 hypv-page セクションには、以下のメトリックが含まれます。 hpi ハイパーバイザーのページ・インの数 (1 秒あたり)。 hpit ハイパーバイザーのページ・インに費やされた平均時間 (ミリ秒)。 pmem 区画の論理メモリーをバッキング中の物理メモリーの量。値の計測の単位はギガバイト (GB) です。 -h フラグを -v フラグと共に指定すると、 -v フラグを使用して表示されるメトリックに加えて、以下のメトリックが表示されます。 Time resolving virtualized partition memory page faults メモリー・ページ・フォルトが解決されるまで待機するために、仮想区画がブロックされている合計時間。時間はミリ秒の細分度で、計測の単位は秒です。 Virtualized partition memory page faults 仮想化区画に記録された仮想区画メモリー・ページ・フォルトの総数。 Number of 4 KB page frames loaned 区画内のハイパーバイザーに貸し出されたメモリーの 4 KB ページの数。 Percentage of partition memory loaned 区画内のハイパーバイザーに貸し出されたメモリーのパーセンテージ。 レポートを幅広モードで表示します。 -w -W kthr セクション内に、追加のフィールド w を表示します。このオプションは、 -I フラグと共に指定した場合にのみ許可されます。

注:

- Active Memory Expansion が使用可能な場合、**vmstat** はメモリーの統計情報を拡張ビュー内で報告します。ただし、環境変数 **AME_MEMVIEW** が **TRUE** に設定されている場合は、メモリーの統計情報は **True** ビューに表示されます。
- AME_MEMVIEW** 環境変数は、**-c** オプションを使用して報告されるメモリーの統計情報に影響を与えません。

例

- ブート以後の統計情報の要約を表示するには、次のコマンドを入力します。

```
vmstat
```
- 2 秒間隔で 5 つの要約を表示するには、次のコマンドを入力します。

```
vmstat 2 5
```
- 論理ディスク **scdisk13** と **scdisk14** の統計情報を含む、ブート以後の統計情報の要約を表示するには、次のコマンドを入力します。

```
vmstat scdisk13 scdisk14
```
- fork** 統計情報を表示するには、次のコマンドを入力します。

```
vmstat -f
```
- さまざまなイベントの回数を表示するには、次のコマンドを入力します。

```
vmstat -s
```
- vmstat** の出力の各列の隣にタイム・スタンプを表示するには、次のコマンドを入力します。

```
vmstat -t
```
- I/O 指向のビューに列の代替セットを表示するには、次のコマンドを入力します。

```
vmstat -I
```

8. 使用可能なすべての VMM 統計情報を表示するには、次のコマンドを入力します。
`vmstat -vs`
9. `alp` および `flp` 列を含むラージ・ページ・セクションを 8 秒間隔で表示するには、次のコマンドを入力します。
`vmstat -l 8`
10. 特定のページ・サイズ (例えば、4 KB) に固有の VMM 統計情報を表示するには、次のコマンドを入力します。
`vmstat -p 4K`
11. システムでサポートされているすべてのページ・サイズについての VMM 統計情報を表示するには、次のコマンドを入力します。
`vmstat -p ALL`

または、次のコマンドを入力します。
`vmstat -p all`
12. 特定のページ・サイズ (この例では、4 KB) についての VMM 統計情報のみを表示するには、次のコマンドを入力します。
`vmstat -P 4K`
13. サポートされているすべてのページ・サイズについて、ページごとの VMM 統計情報の内訳のみを表示するには、次のコマンドを入力します。
`vmstat -P ALL`

または、次のコマンドを入力します。
`vmstat -P all`
14. ブート以後のすべての `workload partitions` の統計情報の要約を表示するには、次のコマンドを入力します。
`vmstat -@ ALL`
15. すべての `workload partitions` の、有効なすべての VMM 統計情報を表示するには、次のコマンドを入力します。
`vmstat -vs -@ ALL`
16. `workload partition` から WPAR とシステム全体の両方の VMM 統計情報を表示するには、次のコマンドを入力します。
`vmstat -@`
17. プロセッサ値を 10 倍してその結果を表示するには、次のコマンドを入力します。
`vmstat -S 1`
18. ハイパーバイザーのページに関する統計情報を表示するには、次のコマンドを入力します。
`vmstat -h`
19. ハイパーバイザーに貸し出されているページについての情報を表示するには、次のコマンドを入力します。
`vmstat -vh`
20. メモリー圧縮の統計情報を表示するには (Active Memory Expansion が使用可能な LPAR 内)、次のコマンドを入力します。
`vmstat -c`
21. メモリー圧縮のページ・サイズごとに特有の統計情報を表示するには (Active Memory Expansion が使用可能な LPAR 内)、次のコマンドを入力します。

```
vmstat -c -P ALL
```

22. **-s** オプションにより表示されている統計情報にメモリー圧縮の情報を付加するには (Active Memory Expansion が使用可能な LPAR 内)、次のコマンドを入力します。

```
vmstat -s -c
```

23. **-v** オプションにより表示されている統計情報にメモリー圧縮の情報を付加するには (Active Memory Expansion が使用可能な LPAR 内)、次のコマンドを入力します。

```
vmstat -v -c
```

ファイル

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

関連資料:

35 ページの『vmo コマンド』

関連情報:

iostat コマンド

メモリー・パフォーマンス

vpdadd コマンド

目的

プロダクト、lpp、ヒストリー、およびベンダー・データベースにエントリーを追加します。

構文

```
vpdadd { -c Component | -p Product | -f Feature } -v v.r.m.f [ -D Destdir ] [ -U Command ] [ -R Prereq ] [ -S Msg_Set ] [ -M Msg_Number ] [ -C Msg_Catalog ] [ -P Parent ] [ -I Description ]
```

説明

vpdadd コマンドは、重要プロダクト・データ (VPD) にリストされるインストーラーによって使用されます。VPD は、プロダクト、lpp、およびヒストリー・データベースで構成されます。インベントリー・データベースへのエントリーは、**sysck** コマンドで追加する必要があります。宛先ディレクトリーおよび **non-installp** アンインストーラーを使用するプロダクトをトラッキングする、新しいベンダー・データベースが組み込まれます。

vpdadd コマンドは、最高レベルで *Product*、次に *Feature*、次に *Component* のツリー構造を使用します。

Component は、最下位のインストール可能ユニットですが、この階層では、*Component* のインストールまたはアンインストールを選択できません。したがって、インストーラーが **vpdadd** コマンドを使用してインストール・データベースを更新する場合、それは固有のツリー表現を参照して、その構造に基づいてエントリーを追加します。1 インストールで 1 つのエントリーを追加するだけの場合、*Product* タイプ (*Component* タイプではない) を追加すると、そのエントリーをアンインストール SMIT インターフェースにリストできます。すべてのエントリーは VPD に作成されますが、*Components* および *Features* は、フィルターによってデフォルトの **lspp** リスト (-Lc) から除外されます。

フラグ

項目	説明
-C <i>Msg_Catalog</i>	<i>Component</i> の翻訳された説明を検索するメッセージ・カタログを指定します。デフォルト (英語) の記述は、 -I フラグで指定します。メッセージ・カタログが標準 <code>NLSPATH</code> がない場合、絶対パス名を指定する必要があります。
-c <i>Component</i>	VPD に追加する <i>Component</i> 名を指定します。エントリーは、固有であれば、追加されるのみです。固有であるとは、異なる宛先ディレクトリーを持つということです。データベース内に同じ <i>Component</i> のインスタンスが既に存在していれば、エントリーは追加されず、エラーが戻されます。この場合は強制インストール (つまり再インストール) が可能です。
-D <i>Destdir</i>	インストール時 (および <code>sysck</code> コマンドによるインベントリー・データベースへの追加時) に、 <i>Component</i> のすべてのファイルに追加するルート (接頭部) パスを指定します。 <i>Component</i> 内のファイルは、相対パス名でリストされるので、ルート・パスは変更可能です。デフォルトの宛先ディレクトリーは <code>/opt</code> です。
-f <i>Feature</i>	VPD に追加する <i>Feature</i> 名を指定します。エントリーは、固有であれば、追加されるのみです。固有であるとは、異なる <code>VRMF</code> または宛先ディレクトリーを持つということです。データベース内に同じ <i>Feature</i> のインスタンスが既に存在していれば、エントリーは追加されませんが、エラーは戻されません。この場合は強制インストール (つまり再インストール) が可能です。
-I <i>Description</i>	<i>Component</i> 、 <i>Feature</i> 、または <i>Product</i> のデフォルトの記述を指定します。記述は二重引用符で囲んで指定する必要があります。記述内では単一引用符を使用できますが、二重引用符はその前に <code>¥</code> を付ける必要があります。
-M <i>Msg_Number</i>	記述のメッセージ番号を指定します。
-P <i>Parent</i>	親ソフトウェア・ユニットを指定します。 <i>Component</i> は、ツリー内の位置に応じて、 <i>Feature</i> または <i>Product</i> のいずれかを親として指定します。
-p <i>Product</i>	VPD に追加する <i>Product</i> 名を指定します。エントリーは、固有であれば、追加されるのみです。固有であるとは、異なる <code>VRMF</code> または宛先ディレクトリーを持つということです。データベース内に同じ <i>Product</i> のインスタンスが既に存在していれば、エントリーは追加されませんが、エラーは戻されません。この場合は強制インストール (つまり再インストール) が可能です。
-R <i>Prereq</i>	<i>Component</i> のインストールに必要な <i>Component</i> (ファイルセット) を指定します。引数は引用符で囲んで指定する必要があります。複数の前提条件を指定するために、このフラグは複数回使用できます。これらはインストール時に (インストーラーによって) 前提条件として扱われますが、循環要件チェーンが起きることを避けるために、 <i>Product</i> データベースでは相互必要条件としてリストされません。
-S <i>Msg_Set</i>	メッセージ・セットを指定します (カタログに複数ある場合)。
-U <i>Command</i>	この <i>Component</i> 用にアンインストーラーを起動する <i>Command</i> を指定します。これは単なるコマンド・パス名でもかまいません。または、グローバル・アンインストーラーがある場合は、パラメータを組み込むこともできます。 <code>geninstall</code> コマンドは、このアンインストーラーを呼び出し、 <code>installp</code> は、この値が VPD 内に設定されていれば、ファイルセットをアンインストールしません。追加される <i>Component</i> 、 <i>Feature</i> 、または <i>Product</i> の <code>VRMF</code> 。
-v <i>v.r.m.f</i>	

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

- 下記の例は、レジストリー・サービスが `vpdadd` を呼び出して、*Foo product* 用の *Component* を追加します。この *Component* には、オペレーティング・システムに固有なものと、GUID としてリストされる 2 つの必要条件があります。

```
vpdadd -c EPL2890198489F -v 1.2.3.0 -R "bos.rte.odm 4.3.3.0" -R "8KDEOKY90245686 1.1.0.0" ¥  
-U /usr/opt/foo/uninstaller.class -p KID892KYLIE25 -I "Foo Database Component"
```

- 新規プロダクト `devices.pci.cool.rte` を VPD に追加するには、下記のように入力します。

```
vpdadd -p devices.pci.cool.rte -v 5.1.0.0 -U /usr/sbin/udisetup
```

ファイル

`/usr/sbin/vpdadd`

関連情報:

RBAC

Trusted AIX

lslpp コマンド

geninstall コマンド

vpddel コマンド

目的

プロダクト、lpp、ヒストリー、およびベンダー・データベースからエントリーを除去します。

構文

```
vpddel { -c Component | -p Product | -f Feature } -v V.R.M.F -D Dest_dir
```

説明

`vpddel` コマンドは、プロダクト、lpp、ヒストリー、およびベンダー・データベースからエントリーを除去します。正しいエントリーを除去するために、`vrmf` および宛先ディレクトリーを指定する必要があります。

フラグ

項目	説明
<code>-c Component</code>	指定した <i>Component</i> を除去します。 <i>Component</i> の除去時には、VRMF も組み込む必要があります。
<code>-D Dest_dir</code>	除去する <i>Component</i> の宛先ディレクトリーを指定します。宛先ディレクトリーが組み込まれていない場合、デフォルトの <code>lopt</code> が使用されます。
<code>-f Feature</code>	ベンダー・データベースから除去する <i>Feature</i> を指定します。
<code>-p Product</code>	ベンダー・データベースから除去する <i>Product</i> を指定します。
<code>-v V.R.M.F</code>	VPD およびベンダー・データベースから削除するコンポーネント の、バージョン、リリース、変更、および修正レベルを指定します。

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

Component EPL2890198489F を、プロダクト、ヒストリー、lpp、およびベンダー・データベースから除去するには、下記のとおり入力します。

```
vpddel -c EPL2890198489F -v 1.2.3.0 -D /usr/lpp/Foo
```

ファイル

`/usr/sbin/vpddel`

関連情報:

lslpp コマンド

RBAC

Trusted AIX

vsdata1st コマンド

目的

vsdata1st - 仮想共用ディスク・サブシステム情報を表示します。

構文

vsdata1st `{-g | -n | -v | -c}`

説明

いくつかの情報の種類のうちの 1 つを標準出力に表示するには、このコマンドを使用します。

vsdata1st コマンドを実行するには、System Management Interface Tool (SMIT) を使用することができます。SMIT を使用するには、次のように入力します。

```
smit list_vsd
```

さらに、参照したい仮想共用ディスク SDR 情報の種類に関するオプションを選択します。

フラグ

vsdata1st の各呼び出しを使用して、次のフラグから 1 つだけを指定できます。

- g** 以下のグローバル・ボリューム・グループ・データが表示されます。
 - global_group_name*、
 - local_group_name*、
 - primary_server_node*、
 - secondary_server_node*。(これは、リカバリー可能仮想共用ディスク・サブシステムでのみ使用可能です。)
 - eio_recovery*
 - recovery*
 - CVSD server_list*
- n** 以下のノード・データを表示します。
 - node_number*、
 - host_name*、
 - adapter_name*、
 - min_buddy_buffer_size*、
 - max_buddy_buffer_size*、

max_buddy_buffers。

-v 以下の定義データを表示します。

vsd_name、
logical_volume_name、
global_group_name、
minor_number。

-c 以下のクラスター情報を表示します。

node_number
cluster_name

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 権限が必要です。

終了状況

0 コマンドが正常終了したことを示します。

非ゼロ

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 **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。

標準出力

現行の RVSD サブシステム実行レベル。

例

1. グローバル・ボリューム・グループ・データを表示するには、次のように入力します。

```
vsdata1st -g
```

システムは下記のようなメッセージを表示します。

VSD Global Volume Group Information

Global Volume Group name	Local VG name	Server Node Numbers			recovery	server_list	vsd_type
		primary	backup	eio_recovery			
gpfs0vgv	gpfs0vg	1	2	0	0	0	VSD
gpfs1vgv	gpfs1vg	2	1	0	0	0	VSD
gpfs3vgv	gpfs3vg	1	0	0	0	1:2	CVSD

2. グローバル・ボリューム・グループ・データを表示するには、次のように入力します。

```
vsdata1st -n
```

システムは下記のようなメッセージを表示します。

VSD Node Information

node number	host_name	VSD adapter	IP packet size	Buddy Buffer		
				minimum size	maximum size	# maxbufs
1	host1	m10	61440	4096	262144	128
2	host2	m10	61440	4096	262144	128

3. グローバル・ボリューム・グループ・データを表示するには、次のように入力します。

```
vsdata1st -v
```

システムは下記のようなメッセージを表示します。

```
VSD Table
VSD name          logical volume  Global Volume Group      minor#  size_in_MB
-----
gpfs0vsd          gpfs0lv        gpfs0gvg                 3      4096
gpfs1vsd          gpfs1lv        gpfs1gvg                 1      4096
gpfs3vsd          gpfs3lv        gpfs3gvg                 4      4096
```

位置

`/opt/rsct/vsd/bin/vsdata1st`

関連情報:

lsvsd コマンド

updatevsdnode コマンド

vsdchgserver コマンド

目的

vsdchgserver - 他のノードに対してサーバー・ノードとして現在機能しているノードから、仮想共用ディスク (複数可) のサーバー機能を切り替えます。

構文

vsdchgserver

```
-g vsd_global_volume_group_name -p primary_node
[-b secondary_node] [ -o EIO_recovery]
```

説明

vsdchgserver コマンドを使用すると、2 次ノードにより継承される、または 2 次ノードから提供される 1 次ノードにより継承される、新規の 1 次ノード上に定義されたグローバル・ボリューム・グループにサービス機能の提供を行うことができます。これにより、実際のディスクと接続されたノードの 1 つとの間のケーブルまたはアダプターが機能していない状態であっても、アプリケーションでは仮想共用ディスクの継続使用が可能となります。

リカバリー可能仮想共用ディスク・サブシステムは、**-g** フラグで指定されたグローバル・ボリューム・グループ内で、現在定義されている 1 次ノードおよび 2 次ノードを反転させるために **vsdchgserver** コマンドを使用する場合に限って、自動的に仮想共用ディスク装置を更新します。

フラグ

-g 特定ノード上で定義されたすべての仮想共用ディスクを表すボリューム・グループに対して、グローバル・ボリューム・グループ名を指定します。

- p グローバル・ボリューム・グループに対する 1 次サーバー・ノード番号を指定します。
- b グローバル・ボリューム・グループに対する 2 次ノード番号を指定します。 **-b** フラグを指定しない場合は、2 次ノード定義は除去されます。
- o EIO エラーが起きたときにリカバリーしない場合は、**0** を、リカバリーする場合は、**1** を指定します。

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 権限が必要です。

終了状況

0 コマンドが正常終了したことを示します。

非ゼロ

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 **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。

標準出力

現行の RVSD サブシステム実行レベル。

例

EIO リカバリーを使用して、グローバル・ボリューム・グループ **node12vg** 用の 1 次サーバー・ノードをノード 1 に、また 2 次ノードをノード 2 に変更するには、次のように入力します。

```
vsdchgserver -g node12vg -p 1 -b 2 -o 1
```

位置

```
/opt/rsct/vsd/bin/vsdchgserver
```

vsdelnode コマンド

目的

ノードまたは一連のノードの仮想共用ディスク情報を除去します。

構文

```
vsdelnode node_number ...
```

説明

このコマンドは、ノードまたは一連のノードに対し、仮想共用ディスクのデータを除去するために使用します。

vsdelnode コマンドは、もはや仮想共用ディスクではない、リスト化されたノードを作成するので、仮想共用ディスクはノードからアクセスされることはありません。このコマンドは、グローバル・ボリューム・グループ用のサーバーであるすべてのノードに対して失敗します。

System Management Interface Tool (SMIT) を使用して、**vsdelnode** コマンドを実行できます。SMIT を使用するには、次のように入力します。

```
smit delete_vsd
```

さらに、「仮想共用ディスク・ノードの除去 (Delete Virtual Shared Disk Node)」情報オプションを選択します。

フラグ

- g** 特定ノード上で定義されたすべての仮想共用ディスクを表すボリューム・グループに対して、グローバル・ボリューム・グループ名を指定します。
- p** グローバル・ボリューム・グループに対する 1 次サーバー・ノード番号を指定します。
- b** グローバル・ボリューム・グループに対する 2 次ノード番号を指定します。**-b** フラグを指定しない場合は、2 次ノード定義は除去されます。
- o** EIO エラーが起きたときにリカバリーしない場合は、**0** を、リカバリーする場合は、**1** を指定します。

パラメーター

node_number

除去した仮想共用ディスク情報を「持つ」ノードのノード番号を指定します。

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

リカバリー可能仮想共用ディスク・サブシステムは、削除中のノード (複数の場合もある) 上で停止しなければなりません。そうでない場合、結果は予測不能です。詳細については、「*RSCT for AIX 5L™ Managing Shared Disks*」を参照してください。

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

ノード **3** および **6** の仮想共用ディスクのノード情報を削除するには、次のように入力します。

```
vsdelnode 3 6
```

位置

/opt/rsct/vsd/bin/vsdelnode

vsdelvg コマンド

目的

vsdelvg - 仮想共用ディスクのグローバル・ボリューム・グループ情報を除去します。

構文

```
vsdelvg [-f] global_group_name ...
```

説明

仮想共用ディスクのグローバル・ボリューム・グループ情報を除去するにはこのコマンドを使用します。任意の仮想共用ディスクがグローバル・ボリューム・グループに定義されている場合、**-f** が指定されていないと、**vsdelvg** コマンドは失敗します。**-f** が指定された場合、該当する仮想共用ディスクは、削除される仮想共用ディスクのノード上に定義された状態で、すべて構成解除されます。

vsdelvg コマンドを実行するには、System Management Interface Tool (SMIT) を使用することができます。SMIT を使用するには、次のように入力します。

```
smit delete_vsd
```

さらに、「仮想共用ディスクのグローバル・ボリューム・グループ情報の除去 (**Delete Virtual Shared Disk Global Volume Group Information**)」オプションを選択します。

フラグ

-f このグローバル・ボリューム・グループ上で定義された仮想共用ディスクを、すべて強制的に除去します。

パラメーター

global_group_name

システムに対してこれ以上グローバルとしたりたくないボリューム・グループを指定します。

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。

終了状況

0 コマンドが正常終了したことを示します。

非ゼロ

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 **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。

標準出力

現行の RVSD サブシステム実行レベル。

例

グローバル・ボリューム・グループ **vg1n1** を除去するには、次のように入力します。

```
vsdelvg vg1n1
```

位置

/opt/rsct/vsd/bin/vsdelvg

関連資料:

78 ページの『vsdvg コマンド』

関連情報:

undefvsd コマンド

vsdnode コマンド

目的

ノードまたは一連のノードに対し、仮想共用ディスク情報を定義します。

構文

vsdnode

node_number... adapter_name min_buddy_buffer_size

max_buddy_buffer_size max_buddy_buffers

vsd_max_ip_msg_size [cluster_name]

説明

このコマンドを使用して、指定された仮想共用ディスクのノードの作成、および仮想共用ディスク稼働パラメーターの割り当てを行います。この情報がすべてのノードで同じであれば、このコマンドの実行は一度で済みます。情報がノードによって異なる場合は、同じ仮想共用ディスク情報を持つそれぞれのノードのブロックごとにこのコマンドを実行します。

System Management Interface Tool (SMIT) を使用して、**vsdnode** コマンドを実行できます。SMIT を使用するには、次のように入力します。

```
smit vsd_data
```

さらに、「仮想共通ディスク・ノード情報 (**virtual shared disk Node Information**)」オプションを選択します。

フラグ

-f このグローバル・ボリューム・グループ上で定義された仮想共用ディスクを、すべて強制的に除去します。

パラメーター

node_number

仮想共用ディスク情報を設定するノードまたは一連のノードを指定します。*node_number* に対して指定する値は、有効な RSCT リモート・ピア・ドメイン・ノード番号と一致させなければなりません。

adapter_name

指定されたノードの仮想共用ディスク通信に使用されるアダプター名を指定します。このアダプター名はノードに定義されていなければなりません。仮想共用ディスク・サポートに関係するノードは、正しく通信できるように完全に接続する必要がありますので、注意してください。仮想共用ディスクのデバイス・ドライバーが SP スイッチを介してデータ要求を送信するように指定するには、**ml0** を使用します。この **ml0** アダプターは、仮想共用ディスクのデバイス・ドライバーが次にロードされる時点で使用されます。

min_buddy_buffer_size

仮想共用ディスクへのリモート要求を満たすためにサーバーが使用する最小のバディ・バッファを指定します。この値は 2 の累乗で、4096 以上でなければなりません。推奨値は 4096 (4 KB) です。512 バイト要求の場合、4 KB は多すぎます。ただし、そのサーバー・ノードでリモート要求がプロセスされる間の短い時間にだけ、バディ・バッファが使用されるように呼び出されます。

max_buddy_buffer_size

キャッシュ化されていないリモート要求を満たすためにサーバーが使用する最大のバディ・バッファを指定します。この値は 2 の累乗で、*min_buddy_buffer_size* 以上でなければなりません。推奨値は 262144 (256 KB) です。この値は、仮想共用ディスクで使用中のアプリケーションの I/O 要求サイズ、および仮想共用ディスク・ソフトウェアで使用されるネットワークに依存します。

max_buddy_buffers

割り当てる *max_buddy_buffer_size* バッファの数を指定します。仮想共用ディスクのデバイス・ドライバーは最初にロードされると 1 つの初期サイズが与えられ、その後、必要に応じて追加スペースを動的に割り当てたり再利用します。推奨値は、2000 256 KB バッファです。

バディ・バッファはサーバー上でのみ使用されます。クライアント・ノード上で *max_buddy_buffers* を 1 に設定することができます。

注: **statvsd** コマンドは、リモート要求がバディ・バッファを待機しながらキューイングしているかを示します。

vsd_max_ip_msg_size

仮想共用ディスクに対する最大メッセージ・サイズを、バイト単位で指定します。この値は、ネットワークの最大伝送単位 (MTU) サイズより大きくてはなりません。推奨値は次のとおりです。

- 61440 (60 KB) (スイッチの場合)
- 8192 (8 KB) (ジャンボ・フレームのイーサネットの場合)
- 1024 (1 KB) (1500 バイト MTU のイーサネットの場合)

cluster_name

同時にアクセスされた共用ディスクにサービスしているサーバー・ノードに対し、クラスター名を指定する必要があります。クラスター名にはユーザーが提供する任意の名前を使用できます。1 つのノードは 1 つのクラスターにのみ属します。例えば、同時アクセス環境において、CVSD 用の 2 つのサーバーには、両方ともに同一のクラスター名を指定しなければなりません。

注: *cluster_name* は、SSA (Serial Storage Architecture) ディスクにのみ必要です。

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

このコマンドで指定されたノードは既にピア・ドメインに属していなければなりません。また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

次の例は、スイッチ・ネットワークおよびノード 1 から 8 の情報を定義しています。

```
vsdnode 1 2 3 4 5 6 7 8 m10 4096 262144 128 61440
```

位置

/opt/rsct/vsd/bin/vsdnode

関連情報:

updatevsdnode コマンド

vsdsklst コマンド

目的

ピア・ドメイン全体にわたって、仮想共用ディスク・サブシステムが使用するディスク・リソースを表示する出力結果を生成します。

構文

```
vsdsklst [-v] [-d] [-a | -n node_number[, node_number2, ...]]
```

説明

ピア・ドメイン全体に渡るディスク使用状況を検査するにはこのコマンドを使用します。

フラグ

- v** ボリューム・グループおよびそれに関連付けられた仮想共用ディスクのディスク使用状況のみを表示します。
- d** ボリューム・グループおよびそれに関連付けられた実際のディスクのディスク使用状況のみを表示します。
- a** システムまたはシステム区画内のすべてのノードについて、指定された情報を表示します。
- n *node_number***
情報が表示される 1 つ以上のノード番号をリストします。

パラメーター

node_number

仮想共用ディスク情報を設定するノードまたは一連のノードを指定します。*node_number* に対して指定する値は、有効な RSCT リモート・ピア・ドメイン・ノード番号と一致させなければなりません。

adapter_name

指定されたノードの仮想共用ディスク通信に使用されるアダプター名を指定します。このアダプター名はノードに定義されていなければなりません。仮想共用ディスク・サポートに関係するノードは、正しく通信できるように完全に接続する必要がありますので、注意してください。仮想共用ディスクのデバイス・ドライバーが SP スイッチを介してデータ要求を送信するように指定するには、**ml0** を使用します。この **ml0** アダプターは、仮想共用ディスクのデバイス・ドライバーが次にロードされる時点で使用されます。

min_buddy_buffer_size

仮想共用ディスクへのリモート要求を満たすためにサーバーが使用する最小のバディ・バッファを指定します。この値は 2 の累乗で、4096 以上でなければなりません。推奨値は 4096 (4 KB) です。512 バイト要求の場合、4 KB は多すぎます。ただし、そのサーバー・ノードでリモート要求がプロセスされる間の短い時間にだけ、バディ・バッファが使用されるように呼び出されます。

max_buddy_buffer_size

キャッシュ化されていないリモート要求を満たすためにサーバーが使用する最大のバディ・バッファを指定します。この値は 2 の累乗で、*min_buddy_buffer_size* 以上でなければなりません。推奨値は 262144 (256 KB) です。この値は、仮想共用ディスクで使用中のアプリケーションの I/O 要求サイズ、および仮想共用ディスク・ソフトウェアで使用されるネットワークに依存します。

max_buddy_buffers

割り当てる *max_buddy_buffer_size* バッファの数を指定します。仮想共用ディスクのデバイス・ドライバーは最初にロードされると 1 つの初期サイズが与えられ、その後、必要に応じて追加スペースを動的に割り当てたり再利用します。推奨値は、2000 256 KB バッファです。

バディ・バッファはサーバー上でのみ使用されます。クライアント・ノード上で *max_buddy_buffers* を 1 に設定することができます。

注: **statvsd** コマンドは、リモート要求がバディ・バッファを待機しながらキューイングしているかを示します。

vsd_max_ip_msg_size

仮想共用ディスクに対する最大メッセージ・サイズを、バイト単位で指定します。この値は、ネットワークの最大伝送単位 (MTU) サイズより大きくてはなりません。推奨値は次のとおりです。

- 61440 (60 KB) (スイッチの場合)
- 8192 (8 KB) (ジャンボ・フレームのイーサネットの場合)
- 1024 (1 KB) (1500 バイト MTU のイーサネットの場合)

cluster_name

同時にアクセスされた共用ディスクにサービスしているサーバー・ノードに対し、クラスター名を指定する必要があります。クラスター名にはユーザーが提供する任意の名前を使用できます。1 つのノードは 1 つのクラスターにのみ属します。例えば、同時アクセス環境において、CVSD 用の 2 つのサーバーには、両方ともに同一のクラスター名を指定しなければなりません。

注: *cluster_name* は、SSA (Serial Storage Architecture) ディスクにのみ必要です。

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

次のコマンドの場合:

```
vsdsklst -dv -a
```

ノード 1 もしくは 2 で定義されたボリューム・グループおよび仮想共用ディスクを持つシステムで以下の情報を表示します。

```
c164n12.ppd.pok.ibm.com: Node Number:2; Node Name:c164n12.ppd.pok.ibm.com
c164n12.ppd.pok.ibm.com: Volume group:rootvg; Partition Size:32; Total:271; Free:168
c164n12.ppd.pok.ibm.com:   Physical Disk:hdisk0; Total:271; Free:168
c164n12.ppd.pok.ibm.com:   Volume group:testvg is not varied on.
c164n12.ppd.pok.ibm.com:   Physical Disk:hdisk5;
c164n12.ppd.pok.ibm.com: Volume group:test1vg; Partition Size:4; Total:537; Free:534
c164n12.ppd.pok.ibm.com:   Physical Disk:hdisk2; Total:537; Free:534
c164n12.ppd.pok.ibm.com:   VSD Name:vsd1n2[testnewlv21n2]; Size:1
c164n12.ppd.pok.ibm.com:   VSD Name:vsd2n2[testlv1n2]; Size:346112.25
c164n12.ppd.pok.ibm.com:   VSD Name:vsd3n2[testlv2n2]; Size:346112.25
c164n12.ppd.pok.ibm.com:   Volume group:vg1 is not varied on.
c164n12.ppd.pok.ibm.com:   Physical Disk:hdisk9;
c164n12.ppd.pok.ibm.com:   Volume group:sharkvg is not varied on.
c164n12.ppd.pok.ibm.com:   Physical Disk:hdisk7;
c164n12.ppd.pok.ibm.com:   Physical Disk:hdisk10;
c164n12.ppd.pok.ibm.com: Volume group:bdhclvg; Partition Size:32; Total:134; Free:102
c164n12.ppd.pok.ibm.com:   Physical Disk:hdisk13; Total:134; Free:102
c164n12.ppd.pok.ibm.com: Volume group:gpfs0vg; Partition Size:8; Total:536; Free:0
c164n12.ppd.pok.ibm.com:   Physical Disk:hdisk12; Total:536; Free:0
c164n12.ppd.pok.ibm.com:   VSD Name:gpfs0vsd[gpfs0lv]; Size:352256.75
c164n12.ppd.pok.ibm.com:   Not allocated physical disks:
c164n12.ppd.pok.ibm.com:   Physical disk:hdisk1
c164n12.ppd.pok.ibm.com:   Physical disk:hdisk3
c164n12.ppd.pok.ibm.com:   Physical disk:hdisk4
c164n12.ppd.pok.ibm.com:   Physical disk:hdisk6
c164n12.ppd.pok.ibm.com:   Physical disk:hdisk11
c164n12.ppd.pok.ibm.com:   Physical disk:hdisk15
c164n11.ppd.pok.ibm.com: Node Number:1; Node Name:c164n11.ppd.pok.ibm.com
c164n11.ppd.pok.ibm.com: Volume group:rootvg; Partition Size:32; Total:271; Free:172
c164n11.ppd.pok.ibm.com:   Physical Disk:hdisk0; Total:271; Free:172
c164n11.ppd.pok.ibm.com: Volume group:bdhclvg; Partition Size:32; Total:134; Free:102
c164n11.ppd.pok.ibm.com:   Physical Disk:hdisk9; Total:134; Free:102
c164n11.ppd.pok.ibm.com:   VSD Name:bdhcvsd1n1[lvbdhcvsd1n1]; Size:45056
c164n11.ppd.pok.ibm.com: Volume group:testvg; Partition Size:16; Total:134; Free:70
c164n11.ppd.pok.ibm.com:   Physical Disk:hdisk13; Total:134; Free:70
c164n11.ppd.pok.ibm.com:   Not allocated physical disks:
c164n11.ppd.pok.ibm.com:   Physical disk:hdisk1
c164n11.ppd.pok.ibm.com:   Physical disk:hdisk2
c164n11.ppd.pok.ibm.com:   Physical disk:hdisk3
```

位置

```
/opt/rsct/vsd/bin/vsdsklst
```

vsvdg コマンド

目的

仮想共用ディスクのグローバル・ボリューム・グループを定義します。

構文

```
vsvdg [-g global_volume_group ] {-l server_list local_group_name | local_group_name primary_node  
[secondary_node [eio_recovery]]}
```

説明

このコマンドは、リカバリー可能仮想共用ディスク・サブシステムで使用されるボリューム・グループを定義するために使用します。これはローカル・ボリューム・グループ名、ローカル・ボリューム・グループ名が常駐するノード、およびクラスター全体に渡って認識されているボリューム・グループに基づいた名前を指定することで実行されます。

vsvdg コマンドを実行するには、System Management Interface Tool (SMIT) を使用することができます。SMIT を使用するには、次のコマンドを入力して、「仮想共用ディスクのグローバル・ボリューム・グループ情報 (Virtual Shared Disk Global Volume Group Information)」オプションを選択します。

```
smit vsd_data
```

フラグ

-g *global_volume_group*

新しいグローバル・ボリューム・グループに対して固有の名前を指定します。この名前はシステム区画全体を通して固有でなければなりません。また、将来のシステム区画化操作で名前の競合が起きないように、SP 全体を通しても固有でなければなりません。推奨される命名規則は **vgxxnyy** です。*yy* はノード番号、*xx* はそのノード上のボリューム・グループの固有の番号を指しています。これが指定されていない場合、ローカル・グループ名がグローバル名に用いられます。名前の長さは 31 文字以内でなければなりません。

-l *server_list*

CVSD 用のサーバーのリストを定義します。複数のサーバーは *global_volume_group* が並行ボリューム・グループであることを示します。

パラメーター

local_group_name

仮想共用ディスクに使用されていることを表したいボリューム・グループの名前を指定します。この名前は、名前が常駐しているホストに対しては通用しません。名前の長さは 15 文字以内でなければなりません。

primary_node

ボリューム・グループが常駐する 1 次サーバーのノード番号を指定します。名前の長さは 31 文字以内でなければなりません。

secondary_node

ボリューム・グループが常駐する 2 次サーバーのノード番号を指定します。名前の長さは 31 文字以内でなければなりません。

eio_recovery

EIO エラーに対するリカバリー可能仮想共用ディスク・サブシステムの応答方法を指定します。
`eio_recovery` が値 1 (デフォルト) で設定される場合、EIO エラーが発生すると、リカバリー可能仮想共用ディスク・システムは現行の 1 次ノードおよび 2 次ノードを反転し、新規の 1 次ノードで再試行を行います。

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。

終了状況

0 コマンドが正常終了したことを示します。

非ゼロ

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. 次の例は、`gpfs1gvg` を仮想共用ディスクのグローバル・ボリューム・グループとして定義しています。これには、`primary_node` としての `node1`、および `secondary_node` としての `node2` からアクセスされるローカル・ボリューム・グループ `gpfs1vg` が含まれています。

```
vsdvg -g gpfs1gvg gpfs1vg 1 2
```

2. 次の例は、`gpfs3gvg` を仮想共用ディスクのグローバル・ボリューム・グループとして定義しています。これには、`node1`、および `node2` から並行アクセスされるローカル・ボリューム・グループ `gpfs3vg` が含まれています。

```
vsdvg -g gpfs3gvg -l 1:2 gpfs3vg
```

位置

`/opt/rsct/vsd/bin/vsdvg`

関連資料:

72 ページの『`vsdelvg` コマンド』

関連情報:

`updatevsdvg` コマンド

vsdvgts コマンド

目的

実際のディスクのボリューム・グループ記述子域 (VGDA) からタイム・スタンプを読み込むことにより、リカバリー可能仮想共用ディスク・サブシステムが使用するタイム・スタンプを更新します。

構文

vsdvgts [-a] [*volgrp*]

説明

リカバリー可能仮想共用ディスク・サブシステムが、ツイン接続のボリューム・グループの変更を判別するために使用するタイム・スタンプを更新するには、このコマンドを使用します。サブシステムが変更を検出すると、リカバリー・スクリプトはボリューム・グループをエクスポートし、次にそのボリューム・グループをインポートします。

このコマンドは、エクスポートやインポート操作が実際には必要でない状態において、リカバリー中にボリューム・グループのエクスポート、インポートを回避するために使用することができます。このコマンドの使用にあたっては特に注意しなければなりません。

フラグ

-a 1 次および 2 次ノード両方のボリューム・グループのタイム・スタンプが更新されるように指定します。このフラグを指定しなければ、タイム・スタンプはローカル・ノードでのみ更新されます。

パラメーター

volgrp ボリューム・グループを指定します。このオペランドが指定されていない場合、このノード上のすべてのボリューム・グループに対するタイム・スタンプが更新されます。

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。

終了状況

- 0 コマンドが正常終了したことを示します。
- 1 プログラムが 1 つ以上のタイム・スタンプの読み取りができなかったことを示します。

制約事項

このコマンドはピア・ドメイン内のオンラインのノードから発行する必要があります。オンラインのピア・ドメインに移動するには **starttrpdomain** コマンドを使用します。既存のピア・ドメイン内で、オンラインの特定ノードに移動するには **starttrpnode** コマンドを使用します。RSCT ピア・ドメインの作成と管理の詳細については、「RSCT: Administration Guide」を参照してください。

標準出力

現行の RVSD サブシステム実行レベル。

例

このノードの場合で、仮想共用ディスクのボリューム・グループ vsdvg1 に関連付けられたタイム・スタンプを更新するには、次のように入力します。

```
vsdvgt
```

s vsdvg1

位置

`/usr/lpp/vsd/bin/vsdvgt`s

関連情報:

`updatevsdvg` コマンド

W

以下の AIX コマンドは、文字 *w* で始まります。

w コマンド

目的

現在のシステム・アクティビティーの要約を出力します。

構文

```
w [ -h ] [ -u ] [ -w ] [ -l | -s [ -X ] ] [ -@ [ WPAR ] ] [ User ]
```

説明

w コマンドは、システム上の現在のアクティビティーの要約を出力します。要約には、下記の内容が含まれます。

項目	説明
WPAR	Workload partition の名前。
User	ログインしているユーザー
tty	ユーザーがオンにしている tty の名前
login@	ユーザーがログインした日時
idle	プログラムを最後に端末から読み込もうとしてからの時間 (分) 注: clogin コマンドを指定して wpar にログインすると、アイドル時間はグローバル端末装置から取り込まれます。
JCPU	すべてのプロセスとその子が端末上で使用したシステム装置時間
PCPU	現在アクティブなプロセスが使用したシステム装置時間
What	現行プロセスの名前と引数

要約のヘッダ行には、現在の日時、システムの始動時間、そのシステムにログインしたユーザー数、および負荷平均が表示されます。負荷平均とは、前述の 1 分、5 分、15 分のインターバルで実行可能なプロセスの数です。

下記の例は、ログイン時間フィールドに使用するさまざまなフォーマットを示しています。

項目	説明
10:25am	ユーザーは 24 時間以内にログインしました。
Tue10am	ユーザーは 24 時間以降 7 日目以内にログインしました。
12Mar91	ユーザーは 7 日前より前にログインしました。

User パラメーターにユーザー名を指定した場合には、出力はそのユーザーに限定されます。

フラグ

項目	説明
-@	workload partition名でタグ付けされたシステム活動状況を印刷します。 <ul style="list-style-type: none"> • WPAR 名を伴わない -@ オプションの指定は、システムでアクティブであるすべての WPAR に加え、グローバル環境を示します。また見出し行はグローバル環境のみの値を示します。 • WPAR 名と共に -@ オプションを指定すると活動状況が示され、また見出し行にはその WPAR のみの値が示されません。 • -@ Global の指定は活動状況を示し、また見出し行はグローバル環境のみの値を示します。 <p>注: -@ オプションの指定がないと、w が実行中であれば、現在の WPAR またはグローバル環境を示します。</p>
-h	見出しの出力を抑制します。
-l	長いフォーマットで要約を出力します。これはデフォルトです。
-s	短いフォーマットで要約を出力します。短いフォーマットでは、tty は省略形にされ、ログイン時間、システム装置時間、およびコマンド引数は省略されます。
-u	日時、最後にシステムを始動してからの時間、ログインしているユーザー数、および実行中のプロセス数を出力します。これはデフォルトです。-w フラグまたは -h フラグを指定せずに -u フラグを指定すると、uptime コマンドと同じ働きをします。
-w	-u フラグと -l フラグを指定した場合と同じ働きをします。これがデフォルトとなります。
-X	各ユーザー名を最初の 8 文字に切り捨てるのではなく、ユーザー名のすべての使用可能な文字を印刷します。ユーザー名も、出力の最後のカラムに移動されます。

ファイル

項目	説明
/etc/utmp	ユーザーのリストが入っています。

関連資料:

102 ページの『who コマンド』

関連情報:

uptime コマンド

wait コマンド

目的

あるプロセス ID が終了するまで待ちます。

構文

```
wait [ ProcessID ... ]
```

説明

wait コマンドは、*ProcessID* 変数で指定したプロセス ID が終了するまで待機します (実行を休止します)。 *ProcessID* 変数を指定しない場合には、**wait** コマンドは呼び出し側のシェルが認識しているすべてのプロセス ID が中絶し、終了状況 0 で終了するまで待機します。 *ProcessID* 変数に認識されていないプロセス ID を指定した場合には、**wait** コマンドはそれらを終了状況 127 で終了した既知のプロセス ID として処理します。 **wait** コマンドは、*ProcessID* 変数で指定した最後のプロセス ID の終了状況で終了します。

フラグ

項目	説明
<code>ProcessID</code>	コマンドのプロセス ID (符号なしの 10 進整数) を指定します。 wait コマンドはそのプロセス ID が終了するまで待機します。

終了状況

1 つ以上のオペランドを指定する場合、すべてのオペランドが中断するか、または呼び出したシェルがすべてのオペランドを認識しなければ、**wait** コマンドの終了状況は、指定した最後のオペランドの状況が判明していれば、指定した最後のオペランドが示すコマンドの終了状況情報と同じになります。シグナルを受信したためプロセスが異常終了すると、128 より大きな値の終了状況が戻され、他のシグナルが生成する終了状況情報と区別ができません。しかし、正確な状況値は指定されません (**kill -l** コマンド・オプションを参照)。その他の場合、**wait** コマンドは下記のいずれかの値で終了します。

項目	説明
<code>0</code>	wait コマンドがオペランドなしで呼び出され、呼び出したシェルが認識するプロセス ID はすべて中断しました。
<code>1</code> から	wait コマンドはエラーを検出しました。
<code>126</code>	
<code>127</code>	指定した最後の <code>ProcessID</code> オペランドによって識別されるコマンドが認識されません。

ファイル

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

関連情報:

`kill` コマンド

`sleep` コマンド

`pause` コマンド

wall コマンド

目的

ログインしているすべてのユーザーにメッセージを書き込みます。

構文

```
wall [ -a ] [ -g Group ][ Message ]
```

説明

wall コマンドは、ログインしているユーザー全員にメッセージを書き込みます。`Message` パラメーターを指定しない場合には、**wall** コマンドは、ファイル終わり文字に達するまで、標準入力からメッセージを読み取ります。その後、そのメッセージはログイン中のすべてのユーザーに送信されます。メッセージの先頭には下記の見出しが付けられます。

```
Broadcast message from
user@node
```

```
(tty) at hh:mm:ss ...
```

`hh:mm:ss` は、メッセージが送信された時刻の時、分、および秒を表します。

他のユーザーがセットアップしている保護機能をオーバーライドするには、root ユーザー権限で操作する必要があります。**wall** コマンドは、一般的に root ユーザーが全ユーザーに、システムのシャットダウンの近いことを通知するために使用されます。

注:

- **wall** コマンドはローカル・ノードにだけメッセージを送信します。
- メッセージにはマルチバイト文字を入れることができます。

フラグ

項目	説明
-a	デフォルトの操作を実行します。このフラグは System V との互換性のために提供されます。これは、メッセージをコンソールおよび疑似端末へブロードキャストします。
-g Group	指定したグループにのみブロードキャストします。

ファイル

項目	説明
/dev/tty	デバイスを指定します。

関連資料:

134 ページの『**write** コマンド』

『**wallevnt** コマンド』

関連情報:

mesg コマンド

wallevnt コマンド

目的

イベントまたはリアーム・イベントを、ログインしているすべてのユーザーにブロードキャストします。

構文

wallevnt [-c] [-h]

説明

wallevnt スクリプトは、イベントやリアーム・イベントが発生すると、現在ホストにログインしているすべてのユーザーへ、イベントまたはリアーム・イベントに関するメッセージをブロードキャストします。イベントまたはリアーム・イベントが発生すると、それらに関する情報が、イベント応答リソース・マネージャーによって生成された環境変数へ、そのマネージャーによってキャプチャーされ、ポストされます。このスクリプトは、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できます。また、他のユーザー定義の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。戻される **wallevnt** スクリプトのメッセージの言語は、ロケール設定によって異なります。

これらのスクリプトが応答アクションとなるような、イベントやリアーム・イベントが発生する際に、ログインしているすべてのユーザーのコンソールに、下記のフォーマットのメッセージが表示されます。

Broadcast message from user@host (tty) at hh:mm:ss...

severity event_type occurred for Condition condition_name
on the resource resource_name of resource_class_name at hh:mm:ss mm/dd/yy
The resource was monitored on node_name and resided on {node_names}.

ERRM 環境変数に関するイベント情報が戻され、これには以下の情報も含まれます。

ローカル時間 (Local Time)

イベントまたはリソース・イベントが監視される時を示します。ERRM が提供する実環境変数は ERRM_TIME です。この値は、表示前に読みやすい形式にローカライズされ、変換されます。

このスクリプトは、環境変数の値をキャプチャーし、**wall** コマンドを使用して、メッセージを現在ログインしているユーザーのコンソールへ書き込みます。

フラグ

- c** **wallevent** が ERRM イベントの **ERRM_VALUE** をブロードキャストするよう指定します。 **-c** フラグを指定すると、**wallevent** は SNMP トラップ・メッセージをブロードキャストします。
- h** スクリプトの使用法ステートメントを標準出力へ書き出します。

パラメーター

log_file

イベント情報のログが書き込まれるファイル名を指定します。*log_file* パラメーターには絶対パスを指定する必要があります。

log_file は循環ログとして使用され、64KB の固定サイズです。*log_file* がいっぱいになると、新規のエントリーは、一番古い既存のエントリーを上書きします。

log_file が既に存在する場合、イベント情報は追加されていきます。*log_file* が存在しない場合は、イベント情報が書き込めるように作成されます。

終了状況

- 0 スクリプトは正常に実行されました。
- 1 スクリプトの実行時にエラーが発生しました。

制約事項

1. このスクリプトは ERRM が実行中のノードで実行する必要があります。
2. **wall** コマンドは、現在ログインしているユーザーのコンソールにメッセージを書き出すために使用します。**wall** コマンドの詳細については、**wall** のマニュアル・ページを参照してください。

標準出力

-h フラグが指定されると、スクリプトの使用法ステートメントが標準出力へ書き込まれます。

例

1. **wallevent** スクリプトは、重大通知応答の事前定義アクションであり、リソース **/var** の **/var** スペース使用 条件に関連があると仮定します。この条件に定義されているイベント式のしきい値を超えると、イベントが発生します。重大通知応答が発生して、**wallevent** が実行されます。下記のメッセージが、ログインしているすべてのユーザーのコンソールに表示されます。

Broadcast message from joe@neverland.com (pts/6) at 18:42:03...

Critical event occurred for Condition /var space used
on the resource /var of fileysys of IBM.FileSystem at 18:41:50 03/28/02
The resource was monitored on c174n05 and resided on {c174n05}.

- リソース **/var** の **/var** スペース使用条件に対し、リアーム・イベントが発生すると、下記のメッセージが、ログインしているすべてのユーザーのコンソールに表示されます。

Broadcast message from joe@neverland.com (pts/6) at 18:42:03...

Critical rearm event occurred for Condition /var space used
on the resource /var of fileysys of IBM.FileSystem at 18:41:50 03/28/02
The resource was monitored on c174n05 and resided on {c174n05}.

位置

/opt/rsct/bin/wallevent

関連資料:

85 ページの『wall コマンド』

watch コマンド

目的

信頼性のないプログラムを監視します。

構文

watch [-e *Events*] [-o *File*] [-X] *Command* [*Parameter ...*]

説明

watch コマンドによって、root ユーザーまたは監査グループのメンバーは、信頼できないと思われるプログラムの動作を監視することができます。**watch** コマンドは、*Parameter* フィールドの有無にかかわらず、*Command* パラメーターで指定されたプログラムを開始し、すべての監査イベントまたは **-e** フラグで指定された監査イベントを記録します。

watch コマンドは、プログラムの実行中に作成されたすべてのプロセス (子プロセスを含む) を監視します。**watch** コマンドは、発生するすべてのイベントを監視するために、作成されたプロセスを含むすべてのプロセスが終了するまで継続します。

watch コマンドは、監査レコードをフォーマットし、標準出力または **-o** フラグで指定されたファイルに書き込みます。

watch コマンドが機能するためには、監査サブシステムが構成され、使用可能になってはなりません。

フラグ

項目	説明
-e Events	監査の対象となるイベントを指定します。Events パラメーターは、 <code>/etc/security/audit/events</code> ファイルに定義された監査イベントをコンマで区切ったリストです。デフォルト値はすべてのイベントです。
-o File	出力ファイルのパス名を指定します。 -o フラグを使用しなければ、出力は標準出力に書き出されません。
-X	ユーザー名を表示する他のフラグを指定して使用する場合、長いユーザー名を出力します。その上限は、定義済み属性 (PdAt) の <code>max_logname</code> オブジェクト・データ・マネージャー (ODM) 属性およびカスタマイズ属性 (CuAt) のオブジェクト・クラスにより決まります。ユーザー名が <code>max_logname</code> 属性より大きい場合は、 <code>max_logname</code> 属性で指定された文字数から 1 を引いた数に切り捨てられます。

セキュリティ

アクセス制御: このコマンドは、`root` ユーザーおよび監査グループのメンバーに実行 (x) アクセス権を与えます。`setuid` コマンドが `root` ユーザーに設定されます。この設定により、他の監査サブシステムのコマンドとファイル、およびトラステッド・コンピューティング・ベース属性にアクセスすることができません。

アクセスされるファイルは下記のとおりです。

モード	ファイル
r	<code>/dev/audit</code>
x	<code>/usr/sbin/auditstream</code>
x	<code>/usr/sbin/auditselect</code>
x	<code>/usr/sbin/auditpr</code>

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. `bar` コマンドでオープンされたすべてのファイルを見るには、下記のように入力します。

```
watch -e FILE_Open /usr/lpp/foo/bar -x
```

このコマンドは、監査装置をオープンし、`/usr/lpp/foo/bar` コマンドを実行します。次に、全レコードを読み取り、`FILE_Open` のイベント・タイプのファイルを選択してフォーマットします。

2. 信頼できない可能性のある `xyzproduct` プログラムのインストールを監視するには、下記のように入力します。

```
watch /usr/sbin/installp xyzproduct
```

このコマンドは、監査装置をオープンし、`/usr/sbin/installp` コマンドを実行します。次に、すべてのレコードを読み取り、それらをフォーマットします。

ファイル

項目	説明
/usr/sbin/watch	watch コマンドが入っています。
/dev/audit	監査レコードを読み取る監査装置を指定します。

関連情報:

audit コマンド

RBAC

Trusted AIX

wc コマンド

目的

ファイル内の行数、ワード数、バイト数、または文字数を数えます。

構文

```
wc [ -c | -m ] [ -l ] [ -w ] [ File ... ]
```

```
wc -k [ -c ] [ -l ] [ -w ] [ File ... ]
```

説明

デフォルトでは、**wc** コマンドは *File* パラメーターで指定したファイル内の行数、ワード数、およびバイト数をカウントします。このコマンドは、改行文字数、ワード数、バイト数を標準出力に書き出し、指定されたすべてのファイルの合計カウントを保持します。

File パラメーターを使用すると、**wc** コマンドはファイル名と要求されたカウントを表示します。*File* パラメーターでファイル名を指定しなければ、**wc** コマンドは標準入力を使用します。

wc コマンドは、**LANG**、**LC_ALL**、**LC_CTYPE**、**LC_MESSAGES** の環境変数によって左右されます。

wc コマンドは、ワードを、ホワイト・スペースで区切られたゼロ以外の長さの文字列 (SPACE、TAB など) と見なします。

フラグ

項目	説明
-c	-k フラグを指定しなければ、バイト数をカウントします。 -k フラグを指定すると、 wc コマンドは文字数をカウントします。
-k	文字数をカウントします。 -k フラグを指定すると、 -klwc フラグを指定した場合と同じ働きをします。 -k を他のフラグと一緒に指定する場合は、 -c フラグも指定する必要があります。そうしないと、 -k フラグは無視されます。詳細については、例 4 と例 5 を参照してください。 注: このフラグは、将来のリリースでは無くなる予定です。
-l	行数をカウントします。
-m	文字数をカウントします。このフラグは -c フラグと一緒に使用することはできません。
-w	ワード数をカウントします。単語は、スペース、タブ、または改行文字で区切られた文字の文字列として定義されません。

注: フラグが指定されていない場合、**wc** は、デフォルトで、ファイル内または標準入力の行数、ワード数、バイト数をカウントします。

終了状況

このコマンドは、下記の終了値を戻します。

項目	説明
0	正常終了しました。
>0	エラーが発生しました。

例

1. ファイルの行、ワード、およびバイトのカウンタを表示するには、以下のように入力します。

```
wc chap1
```

wc コマンドは `chap1` ファイルの行、ワード、およびバイトの数を表示します。

2. バイトおよびワードのカウンタだけを表示するには、以下のように入力します。

```
wc -cw chap*
```

wc コマンドは、名前が `chap` で始まるファイル内のバイト数とワード数を表示します。また、これらのファイル内の合計バイト数と合計ワード数も表示します。

3. ファイルの行、ワード、および文字のカウンタを表示するには、以下のように入力します。

```
wc -k chap1
```

wc コマンドは、`chap1` ファイル内の行数、ワード数、および文字数を表示します。

4. ファイルのワードおよび文字のカウンタを表示するには、以下のように入力します。

```
wc -kcw chap1
```

wc コマンドは、`chap1` ファイル内の文字数とワード数を表示します。

5. **wc** コマンドを標準入力上で使用するには、以下のように入力します。

```
wc -klw
```

wc コマンドは、標準入力の行数とワード数を表示します。**-k** フラグは無視されます。

6. ファイルの文字数を表示するには、以下のように入力します。

```
wc -m chap1
```

wc コマンドは、`chap1` ファイル内の文字数を表示します。

7. **wc** コマンドを標準入力上で使用するには、以下のように入力します。

```
wc -mlw
```

wc コマンドは、標準入力内の行数、ワード数、および文字数を表示します。

ファイル

項目	説明
/usr/bin/wc, /bin/wc	wc コマンドが入っています。
/usr/ucb/wc	wc コマンドへのシンボリック・リンクが入っています。

関連情報:

ファイル

入出力ディレクトリ

what コマンド

目的

ファイル内の識別情報を表示します。

構文

what [-s] *Pathname/File...*

説明

get コマンドが **@(#)** キー文字を置換するパターンの発生すべてを、**what** コマンドは指定したファイル内で検索します (識別キーワードの説明については、**get** または **prs** コマンドを参照してください)。規則では、置き換えられる値は "**@ (#)**" (二重引用符、アットマーク、左小括弧、ポンド記号、右小括弧、二重引用符) です。ファイルを指定しない場合には、**what** コマンドは標準入力から読み取ります。

what コマンドは、最初の二重引用符 ("), より大記号 (>), 改行文字、円記号 (¥), または null 文字の手前までのデータを標準出力に書き出します。

what コマンドは **get** コマンドと一緒に使用する必要があります。get コマンドは自動的に識別情報を挿入します。また、情報を手作業で挿入するファイル上でも **what** コマンドを使用することができます。

what コマンドは、コンパイラーが挿入したコマンド・ライン・オプションをバイナリー・ファイル内に収めます。AIX コンパイラーによりバイナリー・ファイル内に保存されたコマンド・ライン・オプションは、円記号 (¥), より大記号 (>), または二重引用符 (") をマクロ定義内に含んでいる場合があります。

what コマンドは、コンパイラーにより保存されるコマンド・ライン・オプションを次のように作成します。

行の先頭にパターン **@(#)** とその後に「opt」が続く場合 (これらの間にブランク・スペース (「 」) があってもなくても)、**what** コマンドは「**¥n**」(行の末尾) まで文字を書き込みます。

以下の例は、

- C コンパイラーおよび FORTRAN AIX コンパイラーの場合、**@(#) opt (...)** を使用する。
- C++ AIX コンパイラーの場合、**@(#) opt (...)** を使用する。

行全体が『**@(#)**』の後に出力されます。

注: **what** コマンドは、実行可能ファイルの SCCS 識別文字列の検索には失敗する場合があります。

フラグ

項目	説明
-s	@(#) パターンの最初に出現したもののみを検索します。

終了状況

このコマンドは、下記の終了値を戻します。

項目	説明
0	一致が見つかりました。
1	それ以外の場合です。

例

ファイル `test.c` に、下記の行を含む C プログラムが入っていると仮定します。

```
char ident[ ] = "@(#)Test Program";
```

`test.c` をコンパイルして、`test.o` を作成するには、下記のコマンドを実行します。

```
what test.c test.o
```

これによって、下記のように表示されます。

```
test.c:
Test Program
test.o:
Test Program
```

注: ファイルが現行ディレクトリーにない場合は、ファイルの絶対パス名 `usr/bin/test.c` および `user/bin/test.o` が必要です。

ファイル

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

関連情報:

その他の SCCS コマンドのリストのセクション

`get` コマンド

`prs` コマンド

whatis コマンド

目的

コマンドが実行する機能を説明します。

構文

```
whatis [ -M PathName ] Command ...
```

説明

whatis コマンドは、*Command* パラメーターで指定したコマンド、システム・コール、ライブラリー関数、またはスペシャル・ファイル名を、**catman -w** コマンドを使用して作成したデータベースから検索します。**whatis** コマンドはマニュアル項目からヘッダー行を表示します。その際、**man** コマンドを発行して追加情報を得ることもできます。

whatis コマンドは、**man -f** コマンドを使用した場合と同じ働きをします。

注: **catman -w** コマンドを使用して HTML ライブラリーから **/usr/share/man/whatis** データベースを構築した場合には、セクション 3 はセクション 2 または 3 と同じになります。セクションの詳細については、**man** コマンドを参照してください。

フラグ

項目	説明
-M <i>PathName</i>	代替検索パスを指定します。検索パスは、 <i>PathName</i> パラメーターで指定します。このパスは、ディレクトリーをコロンで区切ったリストで、標準マニュアル・サブディレクトリーは whatis コマンドを使ってディレクトリー内で探されます。

例

ls コマンドの実行内容を調べるには、下記のように入力します。

```
whatis ls
```

これにより、下記の出力が生成されます。

```
ls(1) -Displays the contents of a directory.
```

ファイル

項目	説明
/usr/share/man/whatis	whatis データベースが入っています。

関連情報:

catman コマンド

ls コマンド

man コマンド

whatnow コマンド

目的

ドラフトの後処理のためのプロンプト・インターフェースを開始します。

構文

```
whatnow [ { -draftfolder folder | -nodraftfolder | file } { -draftmessage message | file } ] [ -editor editor | -noedit ] [ -prompt string ]
```

説明

whatnow コマンドは、メッセージの後処理用のインターフェースを提供します。デフォルトでは、インターフェースは現行ドラフト・メッセージ上で操作されます。**whatnow** コマンドを入力すると、ユーザーはこのインターフェースに入ることができ、システムは下記のプロンプトを戻します。

「What now?」

インターフェース内では、**whatnow** サブコマンドを使用してメッセージ・ドラフトを操作できます。サブコマンドのリストを表示するには、「What now?」プロンプトが表示されたら Enter キーを押します。インターフェースを終了するには、q を押します。

-draftfolder フラグを指定していない場合または **\$HOME/.mh_profile** ファイル内の **Draft-Folder:** エントリが未定義である場合には、**whatnow** コマンドは MH ディレクトリを検索して **draft** ファイルを見つけます。**-draftfolder +folder** フラグの後にメッセージを指定すると、**-draftmessage** フラグを指定した場合と同じになります。

whatnow コマンド用のデフォルト・エディターを変更するには、**-editor** フラグを使用するか、または **Editor:** エントリを **UserMhDirectory/.mh_profile** ファイル内に定義してください。

注: **comp** コマンド、**dist** コマンド、**forw** コマンド、**repl** コマンドは、**whatnow** コマンドと同じインターフェースを使用します。

フラグ

項目	説明
-draftfolder +folder	メッセージの入っているフォルダーを指定します。デフォルトでは、システムは UserMhDirectory/draft ファイルを使用します。 -draftfolder +folder フラグの後にメッセージを指定すると、 -draftmessage フラグを使用した場合と同じになります。
-draftmessage message	ドラフト・メッセージを指定します。
-editor editor	editor 変数の値がメッセージの構成または改訂用の初期エディターであることを指定します。
-help	コマンド構文、使用可能なスイッチ (トグル)、バージョン情報をリストします。 注: MH の場合、このフラグの名前のスペルは完全に入力しなければなりません。
<i>file</i>	ユーザーが選択したドラフト・ファイル。
<i>message</i>	メッセージを指定します。メッセージを指定するには、下記の参照情報を使用してください。
<i>Number</i>	メッセージ番号。
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	現行メッセージの直前のメッセージ。
-nodraftfolder	UserMhDirectory/draft ファイルにドラフトを入れます。
-noedit	初期編集を抑制します。
-prompt string	指定した文字列をプロンプトとして使用します。デフォルトの文字列は What now? です。

whatnow サブコマンド

whatnow サブコマンドを使用すると、メッセージの編集、メッセージの後処理の制御、または **whatnow** コマンドの処理の終了を行うことができます。

項目	説明
display [<i>flags</i>]	再配布または応答されるメッセージを表示します。リスト・プログラムに有効な <i>flags</i> パラメーターを指定することができます。(デフォルト・リスト・プログラムを設定するには、 \$HOME/mh_profile ファイル内の <code>lproc:</code> エントリーを使用します。)リスト・プログラムに無効なフラグを指定した場合には、 whatnow コマンドはドラフトのパス名を渡しません。
edit [<i>commandstring</i>]	<i>commandstring</i> パラメーターでメッセージ用のエディターを指定します。エディターおよびそのエディターに有効なフラグを指定することができます。エディターを指定しない場合は、 whatnow コマンドは、 UserMhDirectory/mh_profile ファイル内の <code>Editor:</code> エントリーで指定したエディターを使用します。 <code>Editor:</code> エントリーが未定義の場合は、 whatnow コマンドは前の編集セッションで使用したエディターを始動します。
list [<i>flags</i>]	ドラフトを表示します。リスト・プログラムに有効な <i>flags</i> パラメーターを指定することができます。(デフォルト・リスト・プログラムを指定するには、 \$HOME/mh_profile ファイル内にデフォルトの <code>lproc:</code> エントリーを設定します。)リスト・プログラムに無効なフラグを指定した場合は、 whatnow コマンドはドラフトのパス名を渡しません。
push [<i>flags</i>]	バックグラウンドでメッセージを送信します。 send コマンドに有効なフラグを指定することができます。
quit [-delete]	whatnow セッションを終了します。 -delete フラグを指定した場合は、 whatnow コマンドはドラフトを削除します。そうしない場合は、 whatnow コマンドはドラフトを保管します。
refile [<i>flags</i>] + <i>folder</i>	指定したフォルダー内にドラフトをファイルし、前に指定したフォーマットの新規ドラフトを提供します。 fileproc として機能するコマンド・サービスに有効な <i>flags</i> パラメーターを指定することができます。(デフォルトの <code>fileproc:</code> エントリーを \$HOME/mh_profile ファイル内に設定することができます。)
send [<i>flags</i>]	メッセージを送信します。 send コマンドに有効なフラグを指定することができます。
whom [<i>flags</i>]	メッセージの送信先のアドレスを表示します。 whom コマンドに有効なフラグを指定することができます。

プロファイル・エントリー

以下のエントリーが、**UserMhDirectory/mh_profile** ファイルに入力されます。

項目	説明
<code>Draft-Folder:</code>	ドラフト用のデフォルト・フォルダーを設定します。
<code>Editor:</code>	デフォルト・エディターを設定します。
<code>fileproc:</code>	メッセージのリファイルに使用するプログラムを指定します。
<code>LastEditor-next:</code>	<code>LastEditor</code> 変数で指定したエディターの終了後に使用するエディターを指定します。
<code>lproc:</code>	メッセージの内容をリストするのに使用するプログラムを指定します。
<code>Path:</code>	UserMhDirectory を指定します。
<code>sendproc:</code>	メッセージの送信に使用するプログラムを指定します。
<code>whomproc:</code>	メッセージの送信先となるユーザーを判別するの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. メッセージに応答するとき元のメッセージを表示するには、「What now?」プロンプトに対して下記のように入力します。

```
display
```

システムは元のメッセージを表示します。**dist** または **repl** コマンド以外のコマンドから **display** サブコマンドを入力すると、表示すべき代替メッセージがないことを示すシステム・メッセージを受け取ります。

2. ドラフト・メッセージを vi エディターで編集するには、「What now?」プロンプトに対して下記のように入力します。

```
edit vi
```

3. ドラフト・メッセージを **.mh_profile** ファイルに指定されているデフォルトのエディターで編集するには、「What now?」プロンプトに対して下記のように入力します。

```
edit
```

4. 作成したドラフト・メッセージの内容をリストするには、「What now?」プロンプトに対して下記のように入力します。

```
list
```

構成中のドラフト・メッセージが表示されます。

5. バックグラウンドでドラフト・メッセージを送信し、即時にシェル・プロンプトを得るには、「What now?」プロンプトに対して下記のように入力します。

```
push
```

ドラフト・メッセージが送信され、即座にシェル・プロンプトを受け取ります。

6. ドラフト・メッセージの作成を中止し、後でメッセージの作成を完了できるようにファイル内に保管するには、「What now?」プロンプトに対して下記のように入力します。

```
quit
```

システムは、以下のようなメッセージで応答します。

```
whatnow: draft left on /home/dale/Mail/draft
```

この例では、ユーザー dale のドラフト・メッセージが **/home/dale/Mail/draft** ファイルに保管されます。

7. ドラフト・メッセージの構成を中止し、そのメッセージを削除するには、「What now?」プロンプトに対して下記のように入力します。

```
quit -delete
```

ドラフト・メッセージが削除されると、シェル・プロンプトが表示されます。

8. 作成中のドラフト・メッセージを送信する前に、ファイルに保存するには、「What now?」プロンプトに対して下記のように入力します。

```
refile +tmp
```

システムは、以下のようなメッセージで応答します。

```
Create folder "home/dale/Mail/tmp"?
```

この例では、yes と答えると、ドラフト・メッセージがユーザー dale のフォルダー tmp にファイルされます。

9. 構成したドラフト・メッセージを送信するには、「What now?」プロンプトに対して下記のように入力します。

```
send
```

メッセージが送信されるとシェル・プロンプトが表示されます。

10. ドラフト・メッセージに含まれる全アドレスがメール送信システムに認識されたことを検証するには、「What now?」プロンプトに対して下記のように入力します。

```
whom
```

システムは、以下のようなメッセージで応答します。

```
jeanne... User unknown  
dale@venus... deliverable
```

この例で、メール送信システムは dale@venus を正確なアドレスとして認識しましたが、jeanne を正確なアドレスとして認識しませんでした。

ファイル

項目	説明
<code>\$HOME/mh_profile</code>	MH ユーザー・プロフィールを指定します。
<code>UserMhDirectory/draft</code>	現行メッセージのドラフトが入っています。
<code>/usr/bin/whatnow</code>	whatnow コマンドが入っています。

関連情報:

RBAC

Trusted AIX

refile コマンド

whereis コマンド

目的

プログラムのソース、バイナリー、またはマニュアルの位置を探します。

構文

```
whereis [ -s ] [ -b ] [ -m ] [ -u ] [ { -S | -B | -M } Directory ... ]... -f ] File ...
```

説明

whereis コマンドは、指定されたファイルのソース、バイナリー、およびマニュアルの各セクションの位置を探し出します。名前を与えると、最初に先行パス名コンポーネントと形式 *.ext* (例えば *.c*) のすべての (単一の) 後書き拡張子が取り除かれます。ソース・コード制御システム (**SCCS** を参照) で生成した接頭部 **s.** も処理されます。次に、コマンドは要求されたプログラムを標準ロケーション・リストから見つけようとしています。

無効なオプションを入力すると、使用方法メッセージが戻ります。その他の場合、診断はありません。

フラグ

-b, -s, -m, -u フラグのいずれかを与えると、**whereis** コマンドは、それぞれバイナリー・セクション、ソース・セクション、マニュアル・セクション、または異常セクションのみ (またはそのうちの 2 つ) を検索します。

項目	説明
-b	ファイルのバイナリー・セクションを検索します。
-m	ファイルのマニュアル・セクションを検索します。
-s	ファイルのソース・セクションを検索します。
-u	異常ファイルを検索します。ファイルは、要求された各タイプのエントリーが 1 つなければ、異常と見なされます。 whereis -m -u * と入力すると、現行ディレクトリー内で、文書の含まれていないファイルの検索が行われます。

-B, -M, -S フラグを使用すると、**whereis** コマンドが検索する場所を変更または制限できます。プログラムは実行速度を上げるために **chdir** サブルーチンを使用するので、**-M, -S** および **-B** フラグのディレクトリー・リストの既定パス名は絶対パス名でなくてはなりません。つまり、**/** (スラッシュ) で始まるパス名を指定しなければなりません。

項目	説明
-B	-b と同様です。ただし、検索対象にディレクトリーを追加します。 whereis コマンドがバイナリーの検索を行う場所を変更または制限します。
-M	-m と同様です。ただし、検索対象にディレクトリーを追加します。 whereis コマンドがマニュアル・セクションの検索を行う場所を変更または制限します。
-S	-s と同様です。ただし、検索対象にディレクトリーを追加します。 whereis コマンドがソースの検索を行う場所を変更または制限します。
-f	最後の -M, -S 、または -B ディレクトリー・リストを終了し、ファイル名の開始シグナルを送信します。

例

/usr/ucb ディレクトリー内のファイルのうち、**/usr/man/man1** ディレクトリー内で文書化されていないもの、あるいは **/usr/src/cmd** ディレクトリー内にソースがないものをすべて探すには、下記のように入力します。

```
cd /usr/ucb
whereis -u -M /usr/man/man1 -S /usr/src/cmd -f *
```

ファイル

項目	説明
/usr/share/man/*	マニュアル・ファイルが入っているディレクトリー。
/sbin, /etc, /usr/{lib,bin,ucb,lpp}	バイナリー・ファイルが入っているディレクトリー。
/usr/src/*	ソース・コード・ファイルが入っているディレクトリー。

関連情報:

chdir コマンド

which コマンド

目的

プログラム・ファイルの位置 (別名とパスを含めて) を見つけます。

構文

which [*Name ...*]

説明

which コマンドはプログラム名のリストを入力として受け取り、その名前をコマンドとして入力すると実行されるファイルを検索します。**which** コマンドは、引数が別名の場合は拡張し、ユーザーのパスに従って検索します。別名とパスは、ユーザーのホーム・ディレクトリーにある **.cshrc** ファイルから取り出されます。**.cshrc** ファイルが存在しない場合、またはパスが **.cshrc** ファイルに定義されていない場合、**which** コマンドはユーザーの環境に定義されたパスを使用します。

名前が複数の単語の別名であったり、引数名に関連する実行可能ファイルがパスに見つからなかった場合、診断が戻ります。

Korn シェルの場合、**whence** コマンドを使えばより詳細な報告が得られます。

例

コマンド名 lookup: に関連付けられる実行可能ファイルを見付けるには、下記のように入力します。

```
which lookup
```

ファイル

項目	説明
\$HOME/.cshrc	別名とパス値のソースが入っています。

関連情報:

Korn シェルまたは POSIX シェルの組み込みコマンド

ssh コマンド

ksh コマンド

which_fileset コマンド

目的

指定したファイル名またはコマンドを、**/usr/lpp/bos/AIX_file_list** ファイルから検索します。

構文

which_fileset [*File*]

説明

which_fileset コマンドは、**/usr/lpp/bos/AIX_file_list** を検索して、指定したファイル名またはコマンド名を見つけ、ファイルまたはコマンドの送出先であるファイルセットの名前を出力します。

/usr/lpp/bos/AIX_file_list ファイルはサイズが大きいため、自動的にインストールされません。このファイルを受け取るには、**bos.content_list** ファイルセットをインストールしなければなりません。

File パラメーターは、コマンド名、絶対パス名、または正規表現の検索パターンにすることができます。

例

1. 出荷された **dbx** コマンドがどのファイルセットに入っているかを表示するには、以下のように入力します。

```
which_fileset dbx
```

画面には以下のように表示されます。

```
/usr/bin/dbx > /usr/ccs/bin/dbx          bos.adt.debug 4.2.1.0  
/usr/ccs/bin/dbx                        bos.adt.debug 4.2.1.0
```

2. **sendmail** 文字列を含むコマンドとパスをすべて表示するには、次のように入力します。

```
which_fileset sendmail.*
```

画面には以下のように表示されます。

```
/usr/ucb/mailq > /usr/sbin/sendmail    bos.compat.links 4.2.0.0  
/usr/ucb/newaliases > /usr/sbin/sendmail bos.compat.links 4.2.0.0  
/usr/lib/nls/msg/Ca_ES/sendmail87.cat  bos.msg.Ca_Es.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/ca_ES/sendmail87.cat  bos.msg.ca_Es.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/cs_CZ/sendmail87.cat  bos.msg.cs_CZ.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/De_DE/sendmail87.cat  bos.msg.De_DE.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/de_DE/sendmail87.cat  bos.msg.de_DE.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/En_US/sendmail87.cat  bos.msg.En_US.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/en_US/sendmail87.cat  bos.msg.en_US.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/Es_ES/sendmail87.cat  bos.msg.Es_ES.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/es_ES/sendmail87.cat  bos.msg.es_ES.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/Fr_FR/sendmail87.cat  bos.msg.Fr_FR.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/fr_FR/sendmail87.cat  bos.msg.fr_FR.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/hu_HU/sendmail87.cat  bos.msg.hu_HU.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/It_IT/sendmail87.cat  bos.msg.It_IT.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/it_IT/sendmail87.cat  bos.msg.it_IT.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/Ja_JP/sendmail87.cat  bos.msg.Ja_JP.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/ja_JP/sendmail87.cat  bos.msg.ja_JP.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/ko_KR/sendmail87.cat  bos.msg.ko_KR.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/pl_PL/sendmail87.cat  bos.msg.pl_PL.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/ru_RU/sendmail87.cat  bos.msg.ru_RU.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/Sv_SE/sendmail87.cat  bos.msg.Sv_SE.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/sv_SE/sendmail87.cat  bos.msg.sv_SE.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/ZH_CN/sendmail87.cat  bos.msg.ZH_CN.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/zh_CN/sendmail87.cat  bos.msg.zh_CN.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/Zh_TW/sendmail87.cat  bos.msg.Zh_TW.net.tcp.client 4.2.0.0  
/usr/lib/nls/msg/zh_TW/sendmail87.cat  bos.msg.zh_TW.net.tcp.client 4.2.0.0  
/etc/sendmail.cf                       bos.net.tcp.client.4.2.1.0  
/usr/lib/sendmail > /usr/sbin/sendmail bos.net.tcp.client.4.2.1.0  
/usr/sbin/mailq > /usr/sbin/sendmail   bos.net.tcp.client.4.2.1.0  
/usr/sbin/newaliases > /usr/sbin/sendmail bos.net.tcp.client.4.2.1.0  
/usr/sbin/sendmail                     bos.net.tcp.client.4.2.1.0
```

3. **/usr/sbin/which_fileset** コマンドの送出先を見付けるには、次のように入力します。

```
which_fileset /usr/bin/which_fileset
```

画面には、次のように表示されます。

```
/usr/sbin/which_fileset    bos.rte.install 4.2.1.0
```

who コマンド

目的

現在ログインしているユーザーを識別します。

構文

```
who [ -a | -b -d -i -l -m -p -q -r -s -t -u -w -A -H -T -X ] [ File ]
```

```
who am { i | I }
```

説明

who コマンドは、ローカル・システム上のすべての現行ユーザーに関する情報を表示します。表示される情報は、ログイン名、tty、ログインの日付と時間です。**who am i** または **who am I** と入力すると、ログイン名、tty、ログインした日付と時間が表示されます。リモート・マシンからログインすると、そのコンピュータのホスト名も表示されます。

who コマンドは、ログインしてからの経過時間、コマンド・インタプリター (シェル) のプロセス ID、ログイン、ログオフ、再始動、システム・クロックに対する変更、および初期化プロセスで生成された他のプロセスも表示できます。

who コマンドの一般出力フォーマットは下記のとおりです。

```
Name [State] Line Time [Activity] [Pid] [Exit] (Hostname)
```

ここで、

項目	説明
Name	ユーザーのログイン名を識別します。
State	誰でもその回線に書き込み可能かどうかを示します (-T フラグを参照)。
Line	/dev ディレクトリで見つかった回線名を識別します。
Time	ユーザーがログインした時刻を表します。
Activity	ユーザー回線上で最後にアクティビティーが発生してからの時間 (時、分) を表します。ここで .(ドット) は、最後の 1 分以内の回線アクティビティーを示します。回線から 24 時間を超えて信号がないか、最後にシステムを始動してから一度も使用されていない場合には、このエントリーには old と印が付けられます。
Pid	ユーザーのログイン・シェルのプロセス ID を識別します。
Term	プロセスの終了状況を識別します (-d フラグのセクションを参照)。終了値の詳細については、wait サブルーチンのセクションまたは /usr/include/sys/signal.h ファイルを参照してください。
Exit	終了したプロセスの終了状況を識別します (-d フラグのセクションを参照)。
Hostname	ユーザーのログイン元のコンピュータ名を示します。

情報を得るために、**who** コマンドは常に /etc/utmp ファイルを検査します。File パラメーターで別のファイルを指定した場合、**who** コマンドは代わりにそのファイルを調べます。通常この新規ファイルは、/var/adm/wtmp ファイルまたは /etc/security/failedlogin ファイルです。

File パラメーターで複数のファイル名を指定すると、最後のファイル名のみが使用されます。

注: このコマンドは、ローカル・ノード上のユーザーだけを識別します。

フラグ

項目	説明
-a	<code>/etc/utmp</code> ファイルまたはすべての情報が入っている指定されたファイルを処理します。 <code>-bdlprtTu</code> フラグを指定した場合と同じです。
-b	最新のシステム始動時刻と日付を示します。
-d	<code>init</code> で再生成されずに期限が切れてしまったすべてのプロセスを表示します。非活動プロセスの終了フィールドが表示され、その中には非活動プロセスの打ち切り値および終了値 (<code>wait</code> によって戻される) が入っています。(このフラグを使用すると、プロセスが終了した理由を、アプリケーションが戻すエラー番号で判別することができません。)
-l	すべてのログイン・プロセスをリストします。
-m	現行端末に関する情報のみを表示します。 <code>who -m</code> コマンドは、 <code>who am i</code> コマンドおよび <code>who am I</code> コマンドと同じです。
-p	現在アクティブで、以前 <code>init</code> によって生成されたアクティブ・プロセスをリストします。
-q	ユーザーの早見リストとローカル・システム上のユーザー数を出力します。
-r	プロセスの現行実行レベルを示します。
-s	名前、回線、および時間のフィールドだけをリストします。このフラグはデフォルトです。したがって、 <code>who</code> コマンドと <code>who -s</code> コマンドは同じです。
-t	<code>date</code> コマンドを使用して <code>root</code> ユーザーがシステム・クロックに対して最後に行った変更を示します。システムのインストール後に <code>date</code> コマンドが実行されていなければ、 <code>who -t</code> コマンドを実行しても出力は生成されません。
-u または -i	各現行ユーザーのユーザー名、 <code>tty</code> 、ログイン時間、回線アクティビティ、およびプロセス ID を表示します。
-A	<code>/etc/utmp</code> ファイル内のすべてのアカウント・エントリーを表示します。これらのエントリーは、 <code>acctwtmp</code> コマンドによって生成されます。
-H	ヘッダー (タイトル) を表示します。
-T または -w	<code>tty</code> の状態を表示し、その <code>tty</code> に書き込めるユーザーを下記のように示します。
	+ 誰でも書き込み可能。
	- <code>root</code> ユーザーまたはそのオーナーのみ書き込み可能。
	? 正しくない回線が検出された。
-X	各ユーザー名を最初の 8 文字に切り捨てるのではなく、ユーザー名のすべての使用可能な文字を印刷します。ユーザー名も、出力の最後のコラムに移動されます。

終了状況

このコマンドは、下記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

- ローカル・システム・ノードを使用しているユーザーに関する情報を表示するには、下記のように入力します。

```
who
```

下記のような情報が表示されます。

```
pts/1      Nov 9 00:20   long_username_greater_than_eight_characters   (localhost)
```

- ユーザー名を表示するには、以下のように入力します。

```
who am i
```

下記のような情報が表示されます。

```
george 1ft/0 Jun 8 08:34
```

- ログイン、ログアウト、システム始動、およびシステム・シャットダウンのヒストリーを表示するには、下記のように入力します。

```
who /var/adm/wtmp
```

下記のような情報が表示されます。

```
hank  lft/0  Jun  8  08:34  (ausnix5)
john  lft/0  Jun  8  08:34  (JIKey)
mary  lft/0  Jun  8  08:22  (machine.austin.ibm)
jan   pts4    Jun  8  09:19  (puff.wisc.edu)
```

- ローカル・システム・ノードの実行レベルを表示するには、次のように入力します。

```
who -r
```

下記のような情報が表示されます。

```
. run-level 2 Jun 8 04:15 2 0 s
```

- 現在アクティブで、以前 `init` によって生成されたアクティブ・プロセスを表示するには、下記のように入力します。

```
who -p
```

下記のような情報が表示されます。

```
srcmstr . Jun 8 04:15 old 2896
cron . Jun 8 04:15 old 4809
uprintfd . Jun 8 04:15 old 5158
```

- `-bdlprtTu` フラグを指定して `/var/adm/wtmp` ファイルを処理するには、下記のように入力します。

```
who -a /var/adm/wtmp
```

下記のような情報が表示されます。

```
.      system boot Jun 19 10:13
.      run-level 2 Jun 19 10:13
.      . Jun 19 10:14 old
.      . Jun 19 10:14 old
.      . Jun 19 10:14 old
rc     - . Jun 19 10:13 old
.      . Jun 19 10:16 old
.      . Jun 19 10:14 old
srcmstr - . Jun 19 10:14 old
rctcpip - . Jun 19 10:14 old
rcdce  - . Jun 19 10:14 old
rccm   - . Jun 19 10:15 old
dceupdt - . Jun 19 10:15 old
rcnfs  - . Jun 19 10:15 old
cron   - . Jun 19 10:16 old
piobe  - . Jun 19 10:16 old
qdaemon - . Jun 19 10:16 old
writesrv - . Jun 19 10:16 old
uprintfd - . Jun 19 10:16 old
.      . Jun 19 10:16 old
LOGIN  - lft0 Jun 19 10:16 old
.      . Jun 19 10:16 old
.      . Jun 19 10:16 old
```

ファイル

項目	説明
<code>/etc/utmp</code>	ユーザーとアカウントの情報が入っています。
<code>/etc/security/failedlogin</code>	すべての無効ログインの履歴が入っています。
<code>/var/adm/wtmp</code>	ファイルが最後に作成されてからのすべてのログインの履歴が入っています。
<code>/usr/include/sys/signal.h</code>	終了値のリストが入っています。

関連情報:

`date` コマンド

`wait` コマンド

whoami コマンド

目的

ユーザーのログイン名を表示します。

構文

whoami

説明

whoami コマンドはログイン名を表示します。コマンド **who** を使用して **am i** を指定するのと違い、**whoami** コマンドは `/etc/utmp` ファイルを調べないので、`root` 権限がある場合にも機能します。

ファイル

項目	説明
<code>/etc/passwd</code>	ユーザー ID が入っています。

関連資料:

102 ページの『`who` コマンド』

whodo コマンド

目的

システムでユーザーが実行中のジョブをリストします。

構文

whodo [`-h`] [`-l`] [`-X`] [*User*]

説明

子プロセスだけでなく、その端末用のすべてのプロセスに関する情報を印刷します。

デフォルトでは、コマンドがアクティブなログ・ユーザーごとに作成した出力には、端末名、ユーザー ID、ログイン日時が含まれます。先頭には、日付、時間、およびマシン名が出力されます。この情報の後に、そのユーザー ID に関連するアクティブ・プロセスの記録が続きます。各記録には、端末名、プロセス ID、使用された CPU 時間 (分と秒) およびプロセス名が含まれます。

フラグ

項目	説明
-h	出力に印刷される見出しを抑制します。
-l	詳細フォームの出力を作成します。システムの現行アクティビティーの要約が印刷されます。要約には、下記の内容が含まれます。 User ログインしているユーザー tty ユーザーがオンにしている tty の名前 login@ ユーザーがログインした日時 idle プログラムを最後に端末から読み込もうとしてからの時間 (分) JCPU すべてのプロセスとその子が端末上で使用したシステム装置時間 PCPU 現在アクティブなプロセスが使用したシステム装置時間 what 現行プロセスの名前とパラメーター。 要約の見出し行には、現在の日時、システムが始動してからの経過時間、およびそのシステムにログインしたユーザー数が表示されます。
-X	各ユーザー名を最初の 8 文字に切り捨てるのではなく、ユーザー名のすべての使用可能な文字を印刷します。ユーザー名も、出力の最後のカラムに移動されます。

パラメーター

項目	説明
<i>User</i>	出力を、 <i>User</i> で指定したユーザーに関連するすべてのセッションのみに制限します。複数のユーザー名を同時に指定することはできません。

終了状況

- 0** コマンドは正常に完了しました。
- >0** エラーが発生しました。

例

- whodo** コマンドが、フラグやパラメーターを指定せずに、ホスト "linguist" で起動されると、出力は以下ようになります。

```
Sun Jul 28 16:27:12 2002
linguist
```

```
lft0 jeffg 8:15
? 4136 0:00 dtlogin
? 3408 4:55 dtsession
? 2072 4:37 dtwm
? 17310 0:00 dtexec
? 20904 5:53 dtterm
pts/0 22454 0:00 ksh
pts/0 4360 0:07 ksh
pts/0 25788 0:00 whodo
? 23672 0:00 dtexec
? 27536 0:00 dtterm
pts/3 21508 0:00 ksh
? 23888 0:00 dtexec
? 24384 2:49 dtterm
pts/2 24616 0:00 ksh
pts/2 25002 0:04 ksh
pts/2 26110 0:00 ksh
? 25276 0:00 dtexec
? 27090 0:31 dtterm
pts/1 24232 0:00 ksh
```

```

pts/1 23316 0:01 ksh
? 12566 4:23 dtfile
? 21458 1:35 dtfile

pts/0 jeffg 8:16
pts/0 22454 0:00 ksh
pts/0 4360 0:07 ksh
pts/0 25788 0:00 whodo

pts/1 jeffg 17:8
pts/1 24232 0:00 ksh
pts/1 23316 0:01 ksh

pts/2 jeffg 17:20
pts/2 24616 0:00 ksh
pts/2 25002 0:04 ksh
pts/2 26110 0:00 ksh

pts/3 root 16:26
pts/3 21508 0:00 ksh

```

2. コマンド **whodo -l** が、ホスト `linguist` 上で、以下の出力を生成します。

```

04:33PM up 20 day(s), 22 hr(s), 51 mins(s) 5 user(s)
User tty login@ idle JCPU PCPU what
jeffg lft0 08Jul02 21day(s) /usr/sbin/getty /de
jeffg pts/0 08Jul02 14:00 7 whodo -l
jeffg pts/1 16Jul02 10day(s) 44 9 /usr/bin/ksh
jeffg pts/2 12Jul02 11 8:39 4 /usr/bin/ksh
root pts/3 04:26PM 7 -ksh

```

3. コマンド **whodo -lX** が、ホスト `kq11` 上で、以下の出力を生成します。

```

12:50AM up 3 day(s), 1 hr(s), 41 mins(s) 4 user(s)
tty login@ idle JCPU PCPU what User
tty0 Wed11PM 2day(s) -ksh root
pts/0 12:12AM tn 0 root
pts/1 12:20AM whodo -lX long_username_greater_than_eight_characters
pts/2 Fri05AM 2day(s) -ksh root

```

ファイル

項目	説明
<code>/usr/sbin/whodo</code>	whodo コマンドが入っています。
<code>/etc/utmp</code>	ユーザーのリストが入っています。

関連資料:

102 ページの『[who コマンド](#)』

whois コマンド

目的

ユーザーをユーザー ID または別名で識別します。

構文

```
whois [ -h HostName ] [ . | ! ] [ * ] Name [ ... ]
```

whois ?

説明

`/usr/bin/whois` コマンドは、ユーザー名ディレクトリーを検索し、ユーザー ID または *Name* パラメーターに指定したニックネームに関する情報を表示します。**whois** コマンドは ARPANET ホスト `internic.net` に接続し、そこでユーザー名データベースを調べて情報を得ようとします。**whois** コマンドは、ARPANET 上のユーザーだけに使用されなければなりません。**whois** コマンドの詳しい情報および最近の変更内容については、RFC 812 を参照してください。

注: 使用中のネットワークが ARPANET のような全国的ネットワークである場合には、ホスト名は `internic.net` としてハードコーディングされます。

Name [...] パラメーターは、ディレクトリー検索の対象となるユーザー ID、ホスト名、ネットワーク・アドレス、またはニックネームを表します。**whois** コマンドは、任意指定の ... (3 つのピリオド) の前の文字列と一致する名前について、ワイルドカード検索を行います。

フラグ

項目	説明
.	<i>Name</i> パラメーターに指定した名前について、名前だけの検索を強制的に行います。
!	<i>Name</i> パラメーターに指定したニックネームまたはハンドル ID に関するヘルプ情報を表示します。
*	グループまたは組織のメンバーシップ・リスト全体を表示します。メンバーが多い場合には、かなりの時間がかかります。
?	ARPANET ホストからのヘルプを要求します。
-h <i>HostName</i>	代替ホスト名を指定します。ARPANET 上のデフォルトのホスト名は <code>internic.net</code> です。 -h <i>HostName</i> フラグを指定すれば、別の主要な ARPANET ユーザー名データベース <code>nic.ddn.mil</code> に接続することができます。

例

1. Smith という名前で ARPANET に登録済みのユーザーに関する情報を表示するには、下記のように入力します。

```
whois Smith
```

2. ハンドル Hobo を使用する ARPANET 登録ユーザーに関する情報を表示するには、下記のように入力します。

```
whois !Hobo
```

3. John Smith という名前の ARPANET 登録ユーザーに関する情報を表示するには、下記のように入力します。

```
whois .Smith, John
```

4. 名前またはハンドルが文字 HEN から始まる ARPANET 登録ユーザーに関して情報を表示するには、下記のように入力します。

```
whois HEN ...
```

5. **whois** コマンドに関するヘルプ情報を得るには、下記のように入力します。

```
whois ?
```

関連資料:

102 ページの『`who` コマンド』

関連情報:

`named.conf` ファイル・フォーマット

whom コマンド

目的

メッセージ・ハンドラー (MH) のアドレスを操作します。

構文

```
whom [ -alias File ... ] [-nocheck | -check ] [ { -draftfolder +Folder | -nodraftholder | File } {  
-draftmessage Message | -draftFile } ]
```

説明

whom コマンドは以下のことを行います。

- メッセージのヘッダーを 1 組のアドレスに展開します。
- メッセージの予定受信側のアドレスをリストします。
- トランスポート・サービスが送信可能なアドレスかどうかを検査します。

注: **whom** コマンドは、送信可能なアドレスとしてリストされた宛先に実際に送信されるかどうかについては保証しません。

メッセージはドラフト・フォルダーまたはファイルに入れることができます。メッセージの保管場所を指定するには、**-draft**、**-draftfolder**、**-draftmessage** のフラグを使用してください。

-draftfolder フラグを指定しない場合、あるいは **\$HOME/.mh_profile** ファイルの **Draft-Folder:** エントリが未定義である場合には、**whom** コマンドは MH ディレクトリーを検索して **-draft** ファイルを見つけます。 **-draftfolder +Folder** フラグの後にメッセージを指定すると、**-draftmessage** フラグを指定した場合と同じになります。

フラグ

項目	説明
-alias File	メール別名を検索するファイルを指定します。デフォルトの場合は、システムは /etc/mh/Mailaliases ファイルを検索します。
-draft	UserMhDirectory/draft ファイルのヘッダー情報 (存在する場合) を使用します。
-draftfolder +Folder	指定したフォルダー内のドラフト・メッセージからのヘッダー情報を使用します。存在しないドラフト・フォルダーを指定した場合には、システムが新規に作成します。
-draftmessage Message	指定したドラフト・メッセージからのヘッダー情報を使用します。
-help	コマンド構文、使用可能なスイッチ (トグル)、バージョン情報をリストします。 注: MH の場合、このフラグの名前のスペルは完全に入力しなければなりません。
Message	メッセージ・ドラフトを指定します。下記のものを使ってメッセージを指定します。 Number メッセージ番号。 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 現行メッセージの直前のメッセージ。
-nodraftfolder	最後に出現した -draftfolder +Folder フラグを取り消します。

注: 他の 2 種類のフラグ、**-check** および **-nocheck** も使用できます。これらのフラグを指定しても、**whom** コマンドの検査方法は変わりません。**-check** フラグおよび **-nocheck** フラグは互換性だけを検査するものです。

プロファイル・エントリー

以下のエントリーが、*UserMhDirectory/mh_profile* ファイルに入力されます。

項目	説明
Draft-Folder:	ドラフト用のデフォルト・フォルダーを設定します。
postproc:	メッセージの通知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

メッセージの予定受信側のアドレスをリストして検査するには、下記のように、それぞれのプロンプトに対してメッセージの宛先および主題を入力します。

```
To: d77@nostromo
Subject: a test
```

再びプロンプトが表示されたら、メッセージのテキストを入力します。

```
-----Enter initial text
test
-----
```

whatnow プロンプトの後で、**whom** コマンドを入力します。

```
whatnow>>> whom
```

メッセージの予定受信側のアドレスが表示されます。

```
lance...
d77@nostromo... deliverable
```

ファイル

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

関連資料:

94 ページの『whatnow コマンド』

関連情報:

RBAC

Trusted AIX

wlmassign コマンド

目的

ワークロード・マネージメント・クラスにプロセスを手動で割り当てるか、以前にプロセスへ手動で割り当てられたものを取り消します。

構文

```
wlmassign [ -s | -S ] [ -u | Class_Name ] [ pid_list ] [ -g pgid_list ]
```

```
wlmassign [ -t { tag [ -i inheritance ] | -r } ] [ pid_list ] [ -g pgid_list ]
```

説明

wlmassign コマンドでは、次のことが行われます。

- プロセス ID (PID) のリストで指定された一連のプロセスまたはプロセス・グループ ID (PGID) を、指定したスーパークラスとサブクラスの両方または一方に割り当てます。これによって自動クラス割り当てまたは前の手動割り当てを上書きします。
- *pid_list* または *pgid_list* で指定したプロセスの前の手動割り当てを取り消します。
- *pid* または *pgid* のリストによって指定された一連のプロセスに、ワークロード・マネージャー (WLM) タグ・プロセス属性を割り当てます。
- *pid* または *pgid* のリストによって指定された一連のプロセスから、WLM タグ・プロセス属性を除去します。

タグに加えて、追加の継承サブオプションを指定することができます。それらのサブオプションは、子プロセスが **fork** サブルーチンまたは **exec** サブルーチンの後で自分の親からタグを継承する必要があるかどうかを WLM に示します。

プロセスが別のプロセスにタグ付けするためには、少なくとも **SIGPRIV** 特権またはより高位の特権が必要です。

WLM タグ割り当ては、以下の条件の 1 つ以上が当てはまるようになるまで有効です。

- **-r** フラグを使用してタグが削除された。
- タグ付けされたプロセスが終了した。
- タグが新しいタグで上書きされた。

継承属性が無効なクラスに属するプロセスに WLM タグが割り当てられる場合、そのプロセスは現在の割り当て規則に従って自動的に再分類され、その再分類の際に新しいタグが考慮されます。現在のプロセス・クラスにクラス継承属性が指定されていない場合、WLM タグが有効です。タグの規則に基づく再分類によってクラス継承属性をオーバーライドするために、**bos.adt.samples** PTF で使用可能な **/usr/samples/kernel/wlmtune** コマンドを使用して WLM の動作を変更することができます。関連するチューナブルは、以下のとおりです。

tag_override_super

規則がプロセス・タグに適合する場合、ルール・ベースの分類に従ってスーパークラス継承はバイパスされることを WLM に示します。デフォルト値は **0** です。

tag_override_sub

規則がプロセス・タグに適合する場合、ルール・ベースの分類に従ってサブクラス継承はバイパスされることを WLM に示します。デフォルト値は **0** です。

タグ割り当てのルール・ベースの分類を有効にするには、WLM 再分類による更新の前に、チューナブル値を設定しておく必要があります。

自動割り当て (継承と規則)、継承、および手動割り当ての間の相互作用について、詳しくは「オペレーティング・システムおよびデバイスの管理」の『ワークロード・マネージメント』で説明しています。

wlmassign コマンドは、PID のリスト、PGDID のリスト、またはその両方を使用してプロセスを指定できます。これらのリストの形式は、以下のとおりです。

pid[,pid[,pid[...]]]

pgid[,pgid[,pgid[...]]]

有効なスーパークラスまたはサブクラスの名前は、ターゲット・プロセスのクラスへの割り当てを手動で実行するように指定する必要があります。ターゲット・クラスがスーパークラスの場合、このスーパークラスのサブクラスの割り当て規則に従って、各プロセスは、指定したスーパークラスのサブクラスの 1 つに割り当てられます。

手動割り当ては、下記の時点まで有効です (およびプロセスも、その手動での割り当てクラスに残りません)。

- プロセスの終了時。
- ワークロード・マネージメント (WLM) の停止時。 WLM が再始動すると、WLM の停止時に有効であった手動割り当ては消失します。
- プロセスが割り当てられているクラスの削除時。
- 新規の手動割り当ては、前の割り当てを上書きします。
- プロセスの手動割り当ては、**-u** フラグを使用すると取り消されます。
- プロセスは **exec()** ルーチン呼び出しします。

有効なスーパークラスまたはサブクラスの名前は、ターゲット・プロセスのクラスへの割り当てを手動で実行するように指定する必要があります。割り当ては、スーパークラス・レベル、サブクラス・レベル、またはその両方で実行、または取り消すことができます。プロセスの手動割り当てを取り消すか、またはプロセスが **exec()** を呼び出すと、そのプロセスは自動的にクラス分けされます。そのプロセスのクラスに継承が使用可能な場合は、そのクラス内に残るか、または、プロセスが割り当て規則に従って再度クラス分けされます。

手動割り当ての場合、下記のようになります。

- **Class_Name** がスーパークラスの名前である場合、リスト内のプロセスはそのスーパークラスに割り当てられます。次に、ターゲット・スーパークラスのサブクラスの割り当て規則を使用して、それぞれのプロセスのサブクラスが決定されます。
- クラス名がサブクラス名 (**supername.subname**) の場合、デフォルトのプロセスは、スーパークラスとサブクラスの両方に割り当てられます。プロセスは、スーパークラスには **-S** フラグを指定するだけで、またはサブクラスには **-s** フラグを指定するだけで割り当てることができます。

```
wlmassign super1.sub2 -S pid1
```

これは下記と等しくなります。

```
wlmassign super1 pid1
```

プロセスをクラスに割り当てるか、または前の手動の割り当てを取り消すには、ユーザーは、プロセスとターゲット・クラスの両方に対する権限を持っている必要があります。これらの制約は以下のように変換します。

- root ユーザーは、任意のプロセスを任意のクラスに割り当てられます。
- 特定のスーパークラスのサブクラスに対する管理特権があるユーザー (つまり、ユーザーまたはグループ名が、スーパークラスの属性 **adminuser** および **admingroup** に指定されたユーザーまたはグループ名と一致する) は、そのスーパークラスのサブクラスのいずれか 1 つからのプロセスを、他のスーパークラスのサブクラスに手動で再割り当てできます。
- ユーザーは、この固有のプロセス (同じ実または実効ユーザー ID) を、手動割り当て特権がある (つまり、ユーザーまたはグループ名が、スーパークラスまたはサブクラスの属性 **authuser** および **authgroup** に指定されたユーザーまたはグループ名と一致する) クラスに割り当てることができます。

これは、手動でプロセスをクラスに割り当てられるユーザー間で、root を最高として、3 レベルの特権を定義します。ユーザーが手動割り当てを変更または終了するには、最後の手動割り当てを実行したユーザーと、少なくとも同じレベルの特権を持っている必要があります。

注: **wlmassign** コマンドは、現在ロードされている WLM 構成を使用して作業します。現在の構成が 1 つのセットで、そのセット内のすべての構成には存在しないクラスへ割り当てが行われると、そのクラスを含まない構成がアクティブになる時、その割り当ては失われます (クラスが削除される)。

フラグ

項目	説明
-g <i>pgid_list</i>	続くリストが PGID のリストであることを示します。
-S	割り当ては、スーパークラス・レベルでのみ実行、または取り消すことができます。このフラグは、形式 <i>supername.subname</i> のサブクラス名を指定して使用します。
-s	割り当ては、サブクラス・レベルでのみ実行、または取り消すことができます。このフラグは、形式 <i>supername.subname</i> のサブクラス名を指定して使用します。
-u	pid_list または pgid_list からプロセスに対する有効な手動割り当てを取り消します。 -s または -S フラグを使用しない場合、これはスーパークラス・レベルとサブクラス・レベルの両方の手動割り当てを取り消します。
-r	指定されたプロセス・リストまたはプロセス・グループ・リストから WLM タグを除去します。
-t <i>tag</i>	指定されたプロセス・リストまたはプロセス・グループ・リストに WLM タグを設定します。
-i <i>inheritance</i>	1 つまたは両方のタグ継承サブオプションをリスト内にコンマで区切って指定します。以下のタグ継承サブオプションを指定することができます。 fork このプロセスの子が fork において親タグを継承しなければならないことを指定します。 exec このプロセスが exec 呼び出し後に自身のタグを保持することを指定します。

関連情報:

lsclass コマンド

rmclass コマンド

ワークロード管理

wlmscheck コマンド

目的

自動割り当て規則を検査し、および/または、指定した一連の属性を持つプロセスが分類されて入れられる先のワークロード・マネージャー・クラスを判別します。

構文

wlmscheck [**-d** *Config*] [**-a** *Attributes*] [**-q**]

説明

wlmcheck コマンドを、引数を指定しないで実行すると、ワークロード・マネージャー (WLM) の状況が示され、下記のような一貫性検査を行うことができます。

- WLM の現在の状態を表示します (実行中/停止中、アクティブ/パッシブ、アクティブな *rsets* 割り当て、有効な制限合計)。
- 最新のロード・エラーを報告する状況ファイルがあれば、それを表示します。'現行' の構成がセットの場合は、そのセット内のすべての構成にこれを適用し、WLM デーモンがログに記録したメッセージが報告されます。
- 属性と割り当てルール・ファイル (複数の場合も) の一貫性を検査します (クラスの存在、ユーザー名およびグループ名の妥当性、アプリケーション・ファイル名の存在など)。

-d Config フラグを指定しないと、'現行' の構成に対して検査が実行されます。

wlmcheck コマンドは、構成セットに適用できます。この場合、上記の検査は、セット自身を検査した後、セットのすべての構成に対して行われます。スーパークラス名は、それらが属している通常の構成を示すために、「*config/superclass(構成/スーパークラス)*」の形式で報告されます。

構成に **-dConfig** を指定すると、'現行' の代わりに、*Config* 構成またはセットに対し検査を実行します。これは、状況ファイルや WLM デーモン・ログのレポートに影響を与えるのではなく、アクティブな構成のみに適用されます。

-a フラグを指定すると、**wlmcheck** は、*Attributes* で指定した属性を持つプロセスが現行または指定した構成 (あるいは構成セット) に対する規則に従って割り当てられるクラスを表示します。 *Attributes* 文字列のフォーマットは、*rules* ファイル内のエントリーとほぼ同じですが、下記の部分が異なります。

- クラス・フィールドが省略されている (実際には **wlmcheck** の出力)。
- 各フィールドは、最大で 1 つの値を持つことができます。排他 (!)、属性のグルーピング (\$)、コンマで区切ったリスト、およびワイルドカードは使用できません。 *type* フィールドの場合、プロセスは型属性に複数の指定可能な値を同時に取ることができるので、AND 演算子 "+" が使用できます。例えば、プロセスは 32 ビットのプロセスおよび呼び出し *plock* にすることも、64 ビットの固定優先順位処理にすることもできます。
- 少なくとも 1 つのフィールドを指定する必要があります (ハイフン (-) 以外の値を指定します)。

さらに、最初の 2 つのフィールドは必須です。他のフィールドは、存在しない場合はデフォルトでハイフン (-) になりますが、これは割り当て規則の対応するフィールドの値が一致することを意味します。属性文字列内の 1 つ以上のフィールドが、存在しないかまたはハイフン (-) として指定されていない場合、文字列は複数の規則に一致する可能性があります。この場合、**wlmcheck** は、可能なすべての一致に対応するすべてのクラスを表示します。

有効な属性文字列の例 :

```
$ wlmcheck -a "- root system /usr/lib/frame/framemaker - -"  
$ wlmcheck -a "- - staff - 32bit+fixed"  
$ wlmcheck -a "- bob"
```

フラグ

項目	説明
-a <i>Attributes</i>	プロセスが入れられるクラスを判別するために、プロセスの分類属性の一連の値を渡します。これは、割り当て規則が正しく、プロセスが期待どおりに分類されていることを確認する手段です。
-d <i>Config</i>	/etc/wlm/current ではなく、 /etc/wlm/Config (時間ベースの構成セットを示す) 内の WLM 属性ファイルを使用します。
-q	WLM の最新の活動化/更新の状況と、WLM デモン (抑止モード) でログされたメッセージの出力を抑止します。

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
classes	クラスの名前と定義が含まれます。
limits	クラスに課されたリソース制限が含まれます。
rules	自動割り当て規則が含まれます。
shares	クラスに割り当てられたリソース共有が含まれます。

関連情報:

RBAC

Trusted AIX

rules コマンド

wlmcntrl コマンド

目的

ワークロード・マネージャーを始動または停止します。

構文

```
wlmcntrl [ [ -a | -c | -p ] [ -T [ class | proc ] [ -g ] [ -d Config_Dir ] [ -o | -q ] ]
```

```
wlmcntrl -u [ -S Superclass | -d Config_Dir ]
```

説明

wlmcntrl コマンドは、ワークロード・マネージャー (WLM) を停止、開始、更新し、その状態を照会します。WLM の開始または更新時には、ターゲット構成の WLM 属性ファイルにはプリプロセスが実行され、データはカーネルにロードされています。WLM は、下記の 2 つの異なるモードで開始できます。

- アクティブ・モード。このモードでは、WLM は、さまざまなクラスのプロセスでの、プロセッサ、メモリー、およびディスク I/O 使用状況をモニターして調整します。
- パッシブ・モード。このモードでは、WLM は、標準オペレーティング・システム・リソース割り当てメカニズムを妨げることなく、リソースの使用状況をモニターするのみです。

アクティブ・モードが、WLM の通常の操作モードです。

クラスとそれらの制限および共用については、それぞれ、**classes**、**limits**、および **shares** ファイルで記述します。自動割り当て規則は、**rules** ファイルから受け取ります。WLM 構成 **Config** のスーパークラスのクラス属性ファイルは、サブディレクトリー **/etc/wlm/Config** に置かれます。構成 **Config** のスーパークラス **Super** のサブクラスのクラス属性ファイルは、**/etc/wlm/Config/Super** に置かれます。このオペレーティング・システムで出荷される標準構成は、**/etc/wlm/standard** にあります。シンボリック・リンク **/etc/wlm/current** が指すディレクトリーにあるものが、現在の構成です。

-d Config_dir フラグを使用しない場合、**wlmcntrl** は、シンボリック・リンク **/etc/wlm/current** が指すディレクトリーにある構成ファイルを使用します。

-d Config_dir フラグを使用すると、**wlmcntrl** は **/etc/wlm/Config_dir** にある構成ファイルを使用して、**/etc/wlm/Config_dir** を指すシンボリック・リンク **/etc/wlm/current** を更新し、**/etc/wlm/Config_dir** を現在の構成に転換します。**/etc/wlm/Config_dir** を現在の構成に転換するには、この方法をお勧めします。

-u フラグを使用して WLM を更新すると、下記のように、**Config_dir** は空文字列が付いた **-d** フラグとして渡すことができます。

```
wlmcntrl -u -d ""
```

これは、クラス定義を再ロードせずに、現在の構成の割り当て規則をリフレッシュ (再ロード) してカーネルに入れるのみです。これは、WLM が前の活動化の時点で、いくつかのアプリケーション・ファイルがアクセス不可であることを検出した場合に役立つことがあります。システム管理者が、規則またはファイルのいずれかの問題を修正した後、このコマンドは、規則だけを再ロードするために使用できます。

WLM 構成 **Config** は、タイム・ベースの構成のセットでもある可能性があります。この場合、サブディレクトリー **/etc/wlm/Config** には属性ファイルは含まれず、構成のリストと、1 週間にそれらを適用する回数が含まれます。属性ファイルは、そのセット内の通常の構成ごとのサブディレクトリーに入っています。そのようなセットを WLM が開始または更新する際、デーモンは、利用可能な構成を変更すると、セットの通常の構成を切り替える責任があります。

注: このコマンドをワークロード・パーティション内で実行することは、サポートされていません。

フラグ

項目	説明
-a	WLM をアクティブ・モードで開始するか、またはパッシブ・モードからアクティブ・モードに切り替えます。 -d 、 -g 、または -T 以外のフラグを指定していない場合は、これがデフォルトになります。
-c	WLM をプロセッサ専用モードで開始するか、または別のモードからプロセッサ専用モードへ切り替えます。このモードでは、WLM はすべてのリソースに責任を負っています。しかし、プロセッサ・リソースのみが規制されます。
-d Config_dir	/etc/wlm/Config_dir を WLM 構成 (classes 、 limits 、 shares 、および rules ファイルを含む)、または、構成セット (構成のリストやそれらが適用される時刻 tanges を含む) の代替ディレクトリーとして使用します。これは、 /etc/wlm/Config_dir を現在の構成にします。WLM をアクティブ、プロセッサ専用またはパッシブ・モードで開始する場合、または WLM を更新する場合に、このフラグは有効です。このフラグは、 -o および -q フラグと組み合わせて使用したり、モードを切り替える (アクティブ、プロセッサ専用、およびパッシブの間) 際には、使用できません。
-g	WLM に潜在的なリソース・セット割り当てを無視するように指示します。これは、すべてのクラスが、制限付きリソース・セットを使用するかどうかに関係なく、システムのすべてのリソース・セットにアクセスすることを意味します。
-o	ワークロード・マネージャーを停止します。
-p	WLM をパッシブ・モードで開始するか、または別のモードからパッシブ・モードに切り替えます。このモードでは、WLM はすべてのリソースに責任を負っています。しかし、どのリソースも規制されません。

説明

WLM の現在の状態を照会します。下記のを戻します。

- 0 WLM はアクティブ・モードで実行中です。
- 1 WLM は開始していません。
- 2 WLM はパッシブ・モードで実行中です。
- 3 WLM は、rset 割り当てなしのアクティブ・モードで実行中です。
- 4 WLM は、rset 割り当てなしのパッシブ・モードで実行中です。
- 5 WLM はプロセッサ専用のアクティブ・モードで実行中です。
- 6 WLM は、rset 割り当てなしのプロセッサ専用のアクティブ・モードで実行中です。
- 16 WLM はアクティブ・モードで実行中で、プロセス合計アカウントリングはオフです。
- 18 WLM はパッシブ・モードで実行中で、プロセス合計アカウントリングはオフです。
- 19 WLM は rset 割り当てなしのアクティブ・モードで実行中で、プロセス合計アカウントリングはオフです。
- 20 WLM は rset 割り当てなしのパッシブ・モードで実行中で、プロセス合計アカウントリングはオフです。
- 21 WLM はプロセッサ専用のアクティブ・モードで実行中で、プロセス合計アカウントリングはオフです。
- 22 WLM は rset 割り当てなしのプロセッサ専用のアクティブ・モードで実行中で、プロセス合計アカウントリングはオフです。
- 32 WLM はアクティブ・モードで実行中で、クラス合計アカウントリングはオフです。
- 34 WLM はパッシブ・モードで実行中で、クラス合計アカウントリングはオフです。
- 35 WLM は rset 割り当てなしのアクティブ・モードで実行中で、クラス合計アカウントリングはオフです。
- 36 WLM は rset 割り当てなしのパッシブ・モードで実行中で、クラス合計アカウントリングはオフです。
- 37 WLM はプロセッサ専用のアクティブ・モードで実行中で、クラス合計アカウントリングはオフです。
- 38 WLM は rset 割り当てなしのプロセッサ専用のアクティブ・モードで実行中で、クラス合計アカウントリングはオフです。
- 48 WLM はアクティブ・モードで実行中で、クラスおよびプロセス合計アカウントリングはオフです。
- 50 WLM はパッシブ・モードで実行中で、クラスおよびプロセス合計アカウントリングはオフです。
- 51 WLM は rset 割り当てなしのアクティブ・モードで実行中で、クラスおよびプロセス合計アカウントリングはオフです。
- 52 WLM は rset 割り当てなしのパッシブ・モードで実行中で、クラスおよびプロセス合計アカウントリングはオフです。
- 53 WLM はプロセッサ専用のアクティブ・モードで実行中で、クラスおよびプロセス合計アカウントリングはオフです。
- 54 WLM は rset 割り当てなしのプロセッサ専用のアクティブ・モードで実行中で、クラスおよびプロセス合計アカウントリングはオフです。

WLM の現在の状態を示すメッセージは、STDOUT に出力されます。

項目	説明
-S Superclass	スーパークラスのサブクラスに制限されている WLM の更新を要求します。このフラグは -u フラグとともに使用します。実行中の構成がタイム・ベースの構成のセットである場合、スーパークラスは「config/Superclass (構成/スーパークラス)」の形式で指定する必要があります。この場合の「構成」はスーパークラスが属するセットの通常構成です。「構成」が現在アクティブなセットの構成である場合は、すぐに変更が有効になり、そうでない場合は、「構成」が次回アクティブにされる時、有効になります。
-T	クラスおよびプロセスの制限合計アカウンティングと規定の両方が使用不可になります。
-T class	クラスのみ制限合計アカウンティングと規定が使用不可になります。
-T proc	プロセスのみ制限合計アカウンティングと規定が使用不可になります。
-u	WLM を更新します。単一の更新操作は、属性を変更することができ、既存のクラスの制限および共有、クラスの追加または除去の、両方または一方を実行します。実行中の構成がセットである場合、この操作によって、セットの説明と、セット内のすべての構成の内容がリフレッシュされます。更新は、root 権限があるユーザーが、代替の構成または構成セットに切り替えるために使用できます。さらに、更新は、スーパークラス管理者が、管理アクセス権限を持つ (-S フラグを使用して) スーパークラスのサブクラスのみを更新するために使用できます。

セキュリティ

アクセス制御: 開始、停止、モード間の切り替え、およびスーパークラスまたは構成セットの更新には、root 権限が必要です。特定のスーパークラスのサブクラスの更新には、admin user または admin group 特権 (スーパークラス管理者) のみが必要です。すべてのユーザーは、WLM の状態を照会できます。

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。

ファイル

項目	説明
classes	クラスの名前と定義が含まれます。
limits	クラスに課されたリソース制限が含まれます。
rules	自動割り当て規則が含まれます。
shares	クラスに割り当てられたリソース共有が含まれます。
description	それぞれの構成の記述テキストが含まれます。
groupings	その構成用の属性値グループが含まれます。

関連情報:

ワークロード管理

rmclass コマンド

wlmstat コマンド

目的

クラス単位のワークロード・マネージャー (WLM) のリソース使用率統計情報を表示します。

構文

```
wlmstat [-l Class | -t Tier] [-S | -s] [-@] [-c ] [-m] [-b] [-B Device] [-T] [-a] [-w] [-v] [Interval] [Count]
```

```
wlmstat [-l Class | -t Tier] [-@] [-c] [-m] [-b] [-u] [Interval] [Count]
```

wlmstat [-l *Class* | -t *Tier*] [-@] [-M] [-S | -s] [-w] [-v] [*Interval*] [*Count*]

説明

wlmstat コマンドは、カーネルから取り出した WLM データ構造の内容を、シンボルによって表示します。*Count* が指定されると、**wlmstat** は *Count* 回ループし、各ブロックが表示されたあと、*Interval* 秒の間スリープします。*Interval* および *Count* が無指定の場合、1 つの出力レポートが生成されます。*Interval* が指定されたが *Count* が指定されない場合、**wlmstat** は、シグナル (SIGINTR、SIGQUIT、および SIGKILL) によって停止されるまで、指定された間隔で連続して結果を出力します。デフォルトでは、**wlmstat** は、各スーパークラスおよびサブクラスごとにすべてのリソースに対して統計情報を表示します。フラグを指定して、リソースのタイプ、層、スーパークラス、またはサブクラスにまで統計情報の焦点を絞り込み、出力フォーマットを変更することができます。

注: **wlmstat** 出力を表示する場合は、以下の事項を考慮する必要があります。

1. AIX 5.3 からは、WLM プロセッサ使用率の値およびプロセス優先順位調整値は、デフォルトでは 1 秒ごとに 10 回更新されます。
2. プロセッサ使用率として表示される値は、最近の 1 秒における現在の瞬間使用率ではなく、最近 *N* 回の測定値の平均です (AIX 5.3 からは、*N* のデフォルト値は 15 です)。
3. 未管理のクラスは、システム割り込み時間の報告と、WLM によって管理されないシステムのすべてのピンされたページ数に対するメモリー使用率のトラッキングのために使用されます。このクラスにプロセスは割り当てられません。

ハード制限が 50% 指定のプロセスは、連続する 2 回の WLM 使用率の更新の間でプロセッサ使用率 50% を超えて使用可能です。1/10 秒ごとに、全プロセスに優先順位が割り当てられ、そのスケジューラ、次いで各スケジュールがすべて、割り当てられた優先順位に基づいて処理されます。あるプロセスでは、ある WLM 更新から次の更新間にプロセスのハード制限より多くのプロセッサ・リソースを受け取ることがあります。

デフォルトでは、各更新のプロセッサ使用率のそれぞれの瞬間的な測定値は、後続の 15 回の測定読み取りの間は保持され、その他の 14 回の測定値による平均をとった後に、**wlmstat** がその値を表示します。このことにより、ある WLM 更新から次の更新間に 50% より大きい使用率が 1 回発生することが原因で、50% より大きい値となる可能性があります。

あるプロセスがそのハード制限に常時達するかまたはこれを超える場合は、そのプロセスの優先順位は著しく低下し、プロセスを実行できなくなります。長期間でみると、プロセスのリソース使用率は、プロセスのハード最大にするか、またはそれ以下になっている必要があります。短期間でみると、**wlmstat** は、プロセスのハード制限を超えた値を使用したプロセスを表示することがあります。bos.adt.samples PTF で使用できる /usr/samples/kernel/wlmtune コマンドを使用して、そのような場合の WLM の動作を変更できます。関連するチューナブル・フラグは次のとおりです。

schedhz

WLM スケジューラがプロセッサに関するクラス使用量および優先順位を再計算する頻度。デフォルトは 10 です。この値を変更すると、WLM の反応が変わります。この値を大きくすると、WLM は、更新をより頻繁に行うようになります。その結果、プロセスが短期間にそのハード制限を超える可能性が少なくなります。当値変更前よりも WLM 処理が多く発生するため、この代償としてオーバーヘッドが増加します。これにより、システム・パフォーマンス全体が影響を受ける可能性もあります。

cpuhist

平均計算で使用する連続したプロセッサ使用の値をサンプリングする回数。デフォルトは 15 です。この値を大きくすると、長期間の平均値をとることになり、報告されるプロセッサ使用率の値がさらに平滑化されます。

各クラスが長期にわたって最大値を超えないよう WLM の即応性を高めるために、**wlmstat** の出力で希望する結果が表示されるまで、**schedhz** を最初に変更してみることをお勧めします。**wlmstat** が同じ時間間隔で平均をとるように、**cpuhist** を変更することもできます。例えば、**schedhz** が 20、**cpuhist** が 15 の場合は、**wlmstat** は 0.75 (15/20) 秒間にわたって平均をとります。そのため、**wlmstat** が 1.5 秒間にわたって引き続き平均をとるように **cpuhist** を 30 に変更できます。

プロセッサの競合がないシステムでは、WLM の限度を順守するために、**wlmstat** には 5 という *Interval* を指定することをお勧めします。

フラグ

項目	説明
-@	workload partition リソース情報を表示します。
-a	サブクラスの使用量を絶対項で表示します。デフォルトでは、サブクラス使用量のパーセンテージはスーパークラス使用量と比較して表示されます。このオプションを使用すると、サブクラス使用量はシステム上で使用可能なリソースの総計と比較して表示されます (これはスーパークラスの場合と同じです)。すべての値は 1% の精度で表示されます。例えば、スーパークラスに 20% のプロセッサ・ターゲットがあり、 -a オプションを付けない wlmstat ではサブクラスのプロセッサ・パーセンテージが 10% であると表示される場合、 -a オプションを付けた wlmstat ではサブクラスのプロセッサ・パーセンテージが 2% と表示されます。
-b	ディスク I/O 統計情報だけを表示します。
-B <i>Device</i>	ディスク I/O デバイスの統計情報を表示します。空文字列 (-B "") を渡すと、そのクラスがアクセスするすべてのディスクについての統計情報を表示します。
-c	プロセッサ統計情報だけを表示します。
-l <i>Class</i>	<i>Class</i> 名の統計を表示します。指定されない場合は、適合するフィールドの要約とともにすべてのクラスを表示します。
-m	物理メモリー統計情報だけを表示します。
-M	実メモリー/仮想メモリー統計情報を表示します。 -M オプションを使用すると、出力に以下の欄が追加されます。 RMSIZ そのクラスに対して使用された実メモリー・サイズ VMSIZ そのクラスに対して使用された仮想メモリー・サイズ RMLIM クラスの实メモリー限度 VMLIM クラスの仮想メモリー限度 LGPGSIZ そのクラスで使用されたラージ・ページ LGPLIM クラスのラージ・ページ限度
	注: この限度が未定義の場合は、 - が RMLIM 、 VMLIM 、および LGPLIM の各フィールドに表示されます。 -M および -w オプションが併用された場合は、 RMSIZ および VMSIZ のフィールドには、実際に使用された値ではなく、これらの属性の上限水準点が入れられます。さらに、 LGPGSIZ および LGPLIM フィールドはオフにされます。
-s	サブクラス統計情報だけを表示します。
-S	スーパークラス統計情報だけを表示します。
-t <i>Tier</i>	指定した <i>Tier</i> の統計情報だけを表示します。

項目

-T

説明

WLM の開始またはクラスの作成のいずれかのうち、後に行われた方 (時点) からの、リソース使用率の合計数を表示します。単位は下記のとおりです。

CPU クラスが使用した合計プロセッサ時間 (ミリ秒)。

MEM 未使用

DKIO アクセスしたすべてのディスク装置に対して、クラスによって送受信された 512 バイト・ブロックの合計数。

-v

詳細モードを指定します。このフラグは、トラブルシューティングを目的としており、いくつかのクラス属性、リソース共有と各制限、および他の WLM パラメーターも表示します。これには、AIX サポート担当者用の内部パラメーター値も含まれます。以下に示すのは、ユーザーの関心事となる場合がある情報です。

列見出し

説明

CLASS クラス名。

tr 層番号 (0 から 9)。

i 継承属性の値: 0 = いいえ、1 = はい。

#pr クラス内のプロセスの数。クラスに割り当てられているプロセスがない (0) 場合、他の列に示される値は無効であることもあります。

CPU クラスのプロセッサ使用率 (%)。

MEM クラスによる物理メモリーの使用率 (%)。

DKIO クラスのディスク入出力帯域幅の使用率 (%)。

sha 共有の数 ('-' は -1 として表される)。

min リソース最小限度 (%)。

smx リソース・ソフト最大限度 (%)。

hmx リソース・ハード最大限度 (%)。

des (目標値): 共有メンバーを使用する WLM が算出するパーセンテージの目標 (ターゲット) (%)。

npg クラスによって所有されるメモリー・ページの数。

他の列は内部使用だけを目的としており、管理者およびエンド・ユーザーには意味はありません。このフォーマットは、リソース・セレクター (-c、-m、または -b) とともに使用することをお勧めします。そうしないと、行が長すぎて、ディスプレイ端末に収まらない場合があります。

-w

メモリーの最高水準点を表示します。これはクラスが、WLM の開始またはクラスの作成のいずれかが最後に実行された時点以降の、メモリー内に保持されていたページの最大数です。

-u

層ごと、および全体の未使用リソースを表示します。

Display

以下に示すフィールドに結果が作表されます。

名前	クラス名
CPU	クラスによって使用された全プロセッサ時間のパーセンテージ。
MEM	クラスによって使用された物理メモリのパーセンテージ。
DKIO	クラスによって使用されたディスク入出力帯域幅のパーセンテージ。この数は、クラスによってアクセスされたすべてのディスク装置上のディスク帯域幅の平均で、通常は意味のあるものではありません。例えば、クラスが 1 つのディスクの 80% の帯域幅を使用し、他の 2 つのディスクの 5% の帯域幅を使用する場合、DKIO 列は 30% を示します。デバイスごとの使用率の詳細を示すには、-B デバイス・オプションを使用します。

例

1. WLM アクティビティの印刷出力を直ちに取得するには、次のように入力します。

```
wlmstat
```

これにより、下記の出力が生成されます。

```

          CLASS CPU MEM DKIO
Unclassified  0  0  0
Unmanaged    0  0  0
Default      0  0  0
Shared       0  0  0
System       0  0  0
class1      12  0  0
class1.Default  4  0  0
class1.Shared  0  0  0
class1.subclass1  4  0  0
class1.subclass2  4  0  0
class2      12  0  0
class2.Default  4  0  0
class2.Shared  0  0  0
class2.subclass1  4  0  0
class2.subclass2  4  0  0

```

2. スーパークラス **class1** のレポートを取得するには、次のように入力します。

```
wlmstat -l class1
```

これにより、下記の出力が生成されます。

```

          CLASS CPU MEM DKIO
class1      12  0  0
class1.Default  4  0  0
class1.Shared  0  0  0
class1.subclass1  4  0  0
class1.subclass2  4  0  0

```

3. 1 分間で 10 秒ごとに更新されるサブクラス **class1.subclass2** のレポートを取得するには、下記のように入力します。

```
wlmstat -l class1.subclass2 10 6
```

これにより、下記の出力が生成されます。

```

          CLASS CPU MEM DKIO
class1.subclass2  4  0  0

```

4. 仮想メモリー/実メモリー統計情報を表示するには、次のように入力します。

```
wlmstat -M
```

これにより、下記の出力が生成されます。

CLASS	RMSIZ	RMLIM	VMSIZ	VMLIM	LGPGSIZ	LGPGLIM
Unmanaged	1024	4096	4096	8192	0	-
Default	0	-	0	-	0	-
Shared	0	-	0	-	0	-
System	23567	50000	819234	1000000	0	-

5. メモリーの上限水準点を表示するには、次のように入力します。

```
wlmstat -M -w
```

これにより、下記の出力が生成されます。

CLASS	RMSIZ	RMLIM	VMSIZ	VMLIM
Unmanaged	1024	4096	4096	8192
Default	0	-	0	-
Shared	0	-	0	-
System	23567	50000	819234	1000000

エラー

WLM が開始されない場合、**wlmstat** によって警告メッセージが発行されます。

関連資料:

115 ページの『**wlmcntrl** コマンド』

wol コマンド

目的

中断モードでネットワークに接続されている 1 つ以上のホストを、Magic Packet を送信してウェイクアップします。

構文

Magic Packet をサブネット上のブロードキャスト・アドレスに送信する場合は、以下のようになります。

```
wol { [ -m MACAddress [ [ -h Host -s SubnetMask ] | -i Interface ] | -f File } [ -v ]
```

Magic Packet をマルチキャスト・アドレスに送信する場合は、以下のようになります。

```
wol { -m MACAddress -M MulticastAddress [ -p Port ] [ -i Interface ] | -f File } [ -v ]
```

説明

wol コマンドは、中断モードでネットワークに接続されている 1 つ以上のホストを、指定されたサブネット上のアドレスまたは複数のアドレスに Magic Packet を送信してウェイクアップします。

-h フラグも **-s** フラグも指定されない場合、**wol** マネージャーは Magic Packet を以下のようにブロードキャストします。

- インターフェース名 (**-i** *Interface*) を指定すると、Magic Packet は指定したインターフェースからブロードキャストされます。
- インターフェース名が指定されない場合、**wol** マネージャーは、マシンにインストールされた各ネットワーク・インターフェースをループします。インターフェースが作動していると、マネージャーはその

インターフェースから Magic Packet をブロードキャストし、マシンのインターフェース・リスト全体のインターフェースを同様に次から次へと移動します。

-f File で指定したファイルには、ウェイクアップする必要があるホストのリストを含めます。このファイルは 1 つ以上の行で構成され、各行には次のフォーマットで情報を含めます。

MacAddress; Hostname/IPaddress; SubnetMask; Multicast; Port; Interface

例えば、次のようなファイルになります。

```
00:20:35:7a:7:89a;      9.41.86.19;      255.255.255.0 ; ; ;
00:04:ac:17:c0:9f ;      obiwana.aoot.austin.ibm.com;      255.255.255. 224; ; ;
00:07:be:4a:2:394; ; ; ; en0
00:06:38:6b:7e:8f ;      ; ; 234.5.6.7;      12345 ;
```

「#」文字で始まる行はコメントであり、無視されます。各行にはトークンが 6 つ含まれ、それぞれが「;」文字で区切られます。MAC アドレスは必須です。その他のトークンはオプションですが、使わないトークンについても「;」文字を使用する必要があります。

フラグ

項目	説明
-i Interface	wol コマンドの実行に使用する、ホスト上のインターフェースを指定します。
-f File	グループ・リストが含まれるファイルの名前を指定します。これで指定したグループのホストをウェイクアップすることができます。
-h Host	ウェイクアップするホストを、ホスト名またはドット文字列表記 IPv4 アドレス (例えば 10.0.0.3) で指定します。
-m MACAddress	ホストの 48 ビット MAC アドレスを、16 進表記で指定します (例えば 00:20:35:7a:78:9a)。
-M MulticastAddress	IPv4 マルチキャスト・アドレスを指定します。
-p Port	マルチキャスト・マシンで使用するポートを指定します。
-s SubnetMask	IPv4 サブネット・マスクをドット文字列表記で指定します (例えば 255.255.255.0)。
-v	詳細モードを指定します。

終了状況

項目	説明
0	コマンドは正常に完了しました。
>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。

位置

/usr/sbin/wol

関連情報:

RBAC

wparerr コマンド

目的

特定のワークロード・パーティション (WPAR) のエラー・メッセージをログに記録します。

構文

グローバル

```
wparerr -w WparName [{"-c <cat file> -s <set no> -n <msg no>"}] -S default_string args
```

WPAR

```
wparerr [-c <cat file> -s <set no> -n <msg no>] -S default_string args
```

説明

wparerr コマンドは、特定の WPAR のエラー・メッセージをログに記録するメカニズムを提供します。それぞれの WPAR は、最大 1 KB のエラー・メッセージを保持することができます。新規のメッセージを記録するのに十分なスペースがあれば、このコマンドによってメッセージが記録されますが、十分なスペースがなければ失敗します。-w オプションを WPAR の内部で使用してはなりません。-S フラグの後のすべては、メッセージの引数として処理されます。

フラグ

項目	説明
-w	ワークロード・パーティションの名前を指定します。そのワークロード・パーティションのメッセージがログに記録されます。
-c	変換に使用されるカタログ・ファイル名を指定します。
-s	カタログ・ファイル内のエラー・メッセージのメッセージ・セット番号を指定します。
-n	エラー・メッセージのメッセージ番号を指定します。
-S	デフォルトのメッセージ文字列を指定します。printf サブルーチンの Format パラメーターと同じ構文に従います。浮動小数点はサポートされません。
args	メッセージに対する引数があれば、それを指定します。

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。許可と特権の詳細については、「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

- WPAR *mywpar* のメッセージをグローバルからログに記録するには、次のように入力します。


```
wparerr -w mywpar -c wparerrs.msg -s 1 -n 12 -S "%s failed with return value %d\n" lswpar -1
```
- コマンドの失敗を WPAR の内部からログに記録するには、次のように入力します。


```
wparerr -c wparerrs.msg -s 1 -n 5 -S "%s application does not allow checkpoint\n" myapplication
```

関連情報:

kwpar_err コマンド

lssecattr コマンド

wpar_log_err コマンド

wpar_print_err コマンド

wparexec コマンド

目的

アプリケーション・workload partitionまたは仕様ファイルを作成します。

構文

```
wparexec [ -a ] [ -c ] [ -1 ] [ -F ] [ -h hostname ] [-H architecture] [-i] [-I attribute=value ...] ... [ -M attribute=value... ] ... [ -N attribute=value... ] ... [ -R attribute=value... ] [ -u userscript ] [ -v ] [ -x ] { -n wparname [ -e existingwpar | -f infile ] [ -o outfile [ -w ] ] | -f infile [ -n wparname ] [ -o outfile [ -w ] ] | -w -o outfile [ -n wparname ] [ -e existingwpar | -f infile ] } [ [--] [ var=value ...] /path/to/command [ arg ... ] ]
```

注:

ロケールに関わらず、ASCII 文字のみが **wparexec** コマンドの引数として許可されます。

これに加えて、WPAR 名に対する以下の制約事項もあります。

- 25 バイトを超えないものとする。
- 空白文字、または以下の記号のいずれも含まないものとする。

```
= : / ! ; ` ' " < > ~ & ( ) * + [ ] , . ^ 0 { } | ¥
```

- 先頭文字を '-' にも '0' にもしないものとする。

説明

wparexec コマンドは、アプリケーション・workload partitionを構築し開始します。あるいは、今後のアプリケーション・workload partitionsの作成を簡素化するための仕様ファイルを作成します。

アプリケーション・workload partitionは、独自のネットワーク構成とリソース制御プロファイルを持つ、隔離された実行環境です。この区画はグローバル・ファイルシステムのスペースを共有しますが、そこで実行されるプロセスは、同じ区画内の他のプロセスだけから確認できます。この隔離された環境では、プロセスのモニター、リソースの収集、アカウントिंगを行うことができます。また、予定したアプリケーションのクラスターのデータを監査することができます。

wparexec コマンドは、この隔離された環境内にある単一のアプリケーションを始動して、モニターします。**wparexec** コマンドは、workload partition内のすべてのプロセスが終了したときのみ、トラッキングされたこのプロセスの戻りコードと同期して戻ります。例えば、トラッキングされたプロセスがデーモンを作成し、0 の戻りコードを返して終了すると、**wparexec** コマンドはデーモンとその子のすべてが終了するまでブロックします。それから、デーモンまたはその子の戻りコードに関係なく、0 の戻りコードを返して終了します。

フラグ

項目	説明
-l	構成のみを作成します。アプリケーション WPAR の構成を作成後、 wparexec コマンドを停止します。WPAR を始動するには、 startwpar コマンドを使用する必要があります。上級者のみが -l オプションを使用することができます。
-a	必要に応じて、競合する静的設定を自動的に解決します。解決可能な設定は、 name 、 hostname 、およびネットワーク構成です。
-c	このworkload partitionに対してチェックポイントを有効にします。このオプションは、追加のチェックポイント・リスタート・ソフトウェアがインストールされて構成済みであるときのみ有効です。このオプションを使用するときは、workload partitionに関連するファイルシステム (例えば -M オプションを使用) はリモートでなければなりません (例えば vfs=nfs)。
-e existingwpar	仕様データのソースとして既存のアプリケーション・workload partitionを使用します。 -e フラグは -f フラグと一緒に使用しないでください。他の wparexec フラグで指定した値は、既存のworkload partitionから得たこれらの値をオーバーライドします。
-f infile	デフォルト値を読み取るよう仕様ファイルに指示します。このフラグを -e フラグと一緒に使用しないでください。他の wparexec フラグで指定した値は、ロード済み仕様ファイルから得たこれらの値をオーバーライドします。
-F	ほとんどのエラー条件を抑止あるいはオーバーライドします。 -F フラグの使用で、 wparexec コマンドは警告付きで実行を続けます。
-h hostname	このworkload partition用のホスト名を指定します。指定しない場合、 wparexec コマンドは workload partition名をホスト名として使用します。
-H architecture	アーキテクチャー互換のワークロード・パーティションを作成します。有効なアーキテクチャーの値は、 pwr4 、 ppc970 、 pwr5 、 pwr6 、および pwr7 です。アーキテクチャーの値は、システムのハードウェアのバージョンより古い、同じでなければなりません。ワークロード・パーティション内の複数のアプリケーションは、特定のアーキテクチャーの最小共通分母を持ちます。ワークロード・パーティションがチェックポイントを作成できる場合、そのワークロード・パーティションは、ワークロード・パーティション・アーキテクチャー以上のハードウェア・レベルであるシステム間でマイグレーションできます。 注: POWER5 プロセッサ・ベースのシステムと BladeCenter JS21 Express システムには相互に互換性がありません。JS21 は POWER5 プロセッサ・ベースのシステムより低いバージョンのプロセッサを使用しますが、POWER5 プロセッサ・ベースのシステム上で JS21 互換の (ppc970) WPAR を作成することはできません。
-i	workload partition用 WPAR 固有の経路指定を使用可能にします。デフォルトで、workload partitionからの発信ネットワーク・トラフィックは、特に以下の方法でグローバル環境から送信されているように経路指定されます。 <ul style="list-style-type: none"> • 同じグローバル・システム上でホストされていた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介して送信されます。 • 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項目は、workload partitionのトラフィックを伝送するために使用されます。 <p>ユーザーが -i フラグを指定することによって WPAR 固有の経路指定を使用可能にすると、workload partitionは発信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。ブロードキャスト、ループバック、およびサブネットの経路を収容するために、workload partitionのネットワーク・アドレスごとに経路指定項目が自動的に作成されます。ネットワーク属性についての詳細は、-N フラグを参照してください。-I フラグを使用して、workload partitionの経路指定テーブルに明示的な追加項目を作成することができます。特に、デフォルトの経路は自動的に作成されないため、-I フラグを使用してデフォルトの経路を構成することができます。</p>
-I attribute=value ...	WPAR 固有のルーティングが有効なときに作成されるエントリーに、ルーティング・テーブル・エントリーを追加します。複数の -I フラグを指定して、複数の経路を構成することができます。 -i フラグの下に記述されているように、 -I フラグを使用すると、自動的に WPAR 固有の経路指定が使用可能になります。
	以下の属性を -I フラグと一緒に指定することができます。 rtdest 属性と rtgateway 属性を指定する必要があります。
rtdest=destination	経路を割り当てる宛先のホストまたはネットワークを識別します。シンボル名または数値アドレスを使用して、この値を指定することができます。デフォルトの経路を指定するためには、キーワード default を使用できます。経路指定 rtdest 属性についての詳細は、 route コマンドの <i>Destination</i> パラメーターを参照してください。
rtgateway=gateway	パケットが送信される宛先のゲートウェイを識別します。シンボル名または数値アドレスを使用して、この値を指定することができます。
rtnetmask=A.B.C.D	宛先アドレスにネットワーク・マスクを指定します。
rtprefixlen=n	宛先の接頭部の長さを、ネットワークのビット数で指定します。この値は正整数でなければなりません。
rttype={net host}	rtdest 属性を、指定されたタイプとして強制的に解釈させます。
rtinterface=if	インターフェース (例えば、en0) を指定して、経路が選択されたときにこのインターフェースを使用してパケットが送信されるように、その経路と関連付けます。

項目	説明
<p>61Vrtfamily =(inet inet6)</p> <p>-M directory=dir [vfs=type] [dev=devicepath] [host=remotehost] [mountopts=mountopts]</p>	<p>アドレス・ファミリーを指定します。 rtfamily フラグのパラメーターについては、route コマンドの「パラメーター」セクションを参照してください。</p> <p>ファイルシステムの依存関係のみを指定します。属性はスペースで分離する必要があります。デフォルトでは、アプリケーション・workload partitionは、グローバル・ファイルシステムとマウントのすべてに対してworkload partitionを作成したユーザーと同じアクセス・レベルを持ちます。 -M フラグは、ファイルシステム名に設定したディレクトリー属性を使用して、追加のファイルシステムを指定します。 -M フラグは複数指定できます。</p> <p>注: マウントおよびディレクトリーはすべて、グローバルなレベルで作成されて使用可能になります。ディスク (vfs=jfs および vfs=jfs2) にもとづくファイルシステムは、アプリケーション・workload partitions用には作成されません。</p> <p>ローカル・ファイルシステムの依存関係は、ディレクトリー属性のみを定義することにより追加できません。しかし、指定されるディレクトリーは /etc/filesystems に実在していなければなりません。</p> <p>workload partitionの作成プロセス中にエラーが発生すると、wparexec コマンドでマウント済みのファイルシステムはアンマウントされます。作成が成功した場合は、ユーザー・アプリケーションの戻り状況に関係なく、ファイルシステムはアンマウントされません。</p> <p>以下は、アプリケーション・workload partitionsの <i>vfs</i> 属性の有効な値です。</p> <p>nos ホスト属性により指定されたシステムの dev 属性で指定されたディレクトリーは、ディレクトリー属性で指定されたロケーションにマウントされます。マウント・ポイントが存在しない場合は、それが作成されます。nfs マウントに適用できる唯一の他の属性として mountopt 属性があり、これは mount コマンドの -o オプションあるいは /etc/filesystems スタンザにおける options 属性に対応します。指定しない場合、デフォルトでマウント・オプションを使用しません。容認できるオプションの値は、mount コマンドの -o オプションに対応したものです。詳しくは、<i>Commands Reference, Volume 3</i>の mount コマンドを参照してください。</p> <p>namefs dev 属性で指定されたグローバル・ディレクトリーは、directory 属性で指定されたディレクトリー上にマウントされます。</p> <p>namefs マウントに適用できる唯一の他の属性として mountopt 属性があります。</p> <p>directory directory 属性で指定されたグローバル・ディレクトリーは、存在しない場合は作成されます。マウントは実行されません。</p> <p>-n wparname 作成するworkload partitionの名前を指定します。名前が提供されないと、名前が生成されます。</p>

項目
-N attribute=value

説明
次のネットワーク構成の属性を指定することを許可します。

- **interface=** *if* or **interface=***namemappedif*
- **address=***A.B.C.D*
- **netmask=***A.B.C.D*
- **broadcast=***A.B.C.D*
- **address6=***S:T:U:V:W:X:Y:Z*
- **prefixlen=***n*

名前がマップされたインターフェースは、`/etc/wpars/devmap` ファイル内で定義されています。名前がマップされたインターフェースとシステム・インターフェース間のマッピングは、以下のように指定することができます。

```
# The comments start with '#'
# Each line contains a pair of name-mapped interface
# and real interface separated by tab or blank spaces.
foo en0
goo en1
soo en2
```

`attribute=value` の対はスペースで分離する必要があります。複数の `-N` フラグを使用して、複数の IP アドレスを構成することができます。少なくとも、`address` または `address6` 属性を指定する必要があります。`wparexec` コマンドは、指定されない他の値をグローバル・システムの設定から収集します。`-N` フラグが指定されないと、`wparexec` コマンドは、`-n` フラグで指定された `workload partition` 名で `gethostbyname` サブルーチンを実行して、`workload partition` の適切な IP アドレスを発見しようとします。グローバル・インターフェースと同じサブネットでアドレスが検出されると、そのインターフェースの設定は、デフォルトのネットワーク・エントリーを作成するために、解決済みの IP アドレスとともに使用されます。

IPv6 ネットワーク構成を定義するには、`-N` フラグとともに、`address6` 属性、`prefixlen` 属性、および `interface` 属性を指定します。

- `address6` 属性は 128 ビット・アドレスです。このアドレスは、コロンで区切られた 8 つの 16 ビット整数で表されます。それぞれの整数は、4 桁の 16 進数字で表されます。先行ゼロは省略でき、連続する 16 ビット整数は 2 つのコロン (アドレスごとに一回) で表すことができます。
 - `prefixlen` 属性は上位ビットの数であり、IPv6 アドレスをマスクし、接頭部を構成するために使用されます。`prefixlen` 属性の値は 0 から 128 の範囲です。それぞれの `-N` フラグは、IPv4 属性または IPv6 属性を受け入れられますが、両方を受け入れることはできません。
- 仕様データを書き込む出力パスとファイル名を示します。この仕様ファイルは、後で `-f` フラグでアプリケーション・`workload partition` を作成するために使用できます。

-o outfile

項目

-R attribute=value

説明

リソース制御属性を指定することを許可します。1つの**-R**フラグのみが指定できます。ほとんどのリソース制御はワークロード・マネージャー (WLM) でサポートされるものと同じです。これらの属性についての説明は、リストされた WLM ページを参照してください。有効な属性を以下に示します。

active リソース制御の定義の保存は許可されますが、非アクティブにされます。この属性は、*yes* または *no* の値をとります。

rset **mkrset** コマンドによって作成されたリソース・セットを使用して、*workload partition*を構成します。**mkrset** コマンドについての詳細は、*Commands Reference, Volume 3*の **mkrset** コマンドを参照してください。

shares_CPU

この*workload partition*で利用できるプロセッサ共有の数。プロセッサ共有についての詳細は、[ファイル参照](#) のワークロード・マネージャーの共有ファイルを参照してください。

CPU このワークロード・パーティションのプロセスに対するプロセッサ制限のパーセンテージ。この属性は次の形式を使用して、しきい値を限定します。

CPU=<m>%-<SM>%,<HM>%

m 値は、下限を表します。*SM* 値は、ソフト最大制限を表します。*HM* 値は、ハード最大制限を表します。制限付きの値についての詳細は、[ファイル参照](#) のワークロード・マネージャーの制限ファイルを参照してください。

shares_memory

この*workload partition*で利用できるメモリー共有の数。メモリー共有についての詳細は、[ファイル参照](#) のワークロード・マネージャー共有ファイルを参照してください。

memory このワークロード・パーティションのプロセスに対するメモリー制限のパーセンテージ。メモリー制限についての詳細は、[ファイル参照](#) のワークロード・マネージャー制限ファイルを参照してください。

procVirtMem

単一のプロセスが利用できる仮想メモリーの最大量。指定の限界を超えるとプロセスは終了します。有効な単位は、メガバイト (M または MB)、ギガバイト (G または GB)、およびテラバイト (T または TB) です。許される下限は 1 M です。指定できる上限は 8796093022207 M、8589934591 G、または 8388607 T です。値が -1 (単位がない) に設定されていると、制限は無効となります。制限値についての詳細は、[ファイル参照](#) のワークロード・マネージャーの制限ファイルを参照してください。

totalVirtMem

WPAR 全体として利用できる仮想メモリーの最大量。指定された限界を超える原因となるプロセスは終了されます。有効な範囲と単位は、**procVirtMem** の場合と同じです。値が '-1' (単位がない) に設定されていると、制限は無効となります。ワークロード・マネージャー制限ファイル [ファイル参照](#) を参照してください。

totalProcesses

この*workload partition*で許可されるプロセスの合計数。許可されるプロセッサ数についての詳細は、[ファイル参照](#) のワークロード・マネージャー制限ファイルを参照してください。

項目

説明

(-R フラグの属性、続き):

totalPTYs=*n*

このworkload partitionで許可される疑似端末の合計数。許可される疑似端末についての詳細は、pty Special File を参照してください。

totalLargePages=*n*

このworkload partitionに許可できるラージ・ページの数。許可されるラージ・ページについての詳細は、Large Pages を参照してください。

pct_msgIDs=*n*%

このworkload partitionで許可されるシステムのメッセージ・キュー ID の最大数のパーセンテージ。許可されるメッセージ・キュー ID 数についての詳細は、Message Queue Kernel Services を参照してください。

pct_semIDs=*n*%

このworkload partitionで許可されるシステムのセマフォ ID の最大数のパーセンテージ。

pct_shmIDs=*n*%

このworkload partitionで許可されるシステムの共用メモリー ID の最大数のパーセンテージ。許可される共用メモリー ID 数についての詳細は、Shared Memory を参照してください。

pct_pinMem=*n*%

このworkload partitionに割り当て可能なシステムの最大ピン・メモリーのパーセンテージ。ピン・メモリーについての詳細は、Support for pinned memory を参照してください。

totalThreads

このworkload partitionで許可されるスレッドの合計数。許可されるスレッド数についての詳細は、「ファイル参照」のワークロード・マネージャー limits File を参照してください。

-u userscript

種々の管理ポイントにおいて、workload partitionコマンドで実行されるよう、ユーザー・スクリプトへのパスを指定します。-u フラグのパラメーターは、スクリプトに渡される追加の引数を含む引用符付きストリングの場合もあります。-u フラグのパラメーターの最初のコンポーネントは、必ず実在の実行可能ファイルへの絶対パスでなければなりません。スクリプトは次のように呼び出されます。

/path/to/userScript<action><WAPR>

action 引数は、以下のように、実行される管理アクションを示します。

- **WPAR_LOAD:** スクリプトは、カーネル構成後で、トラッキングされたプロセスが作成される前に、グローバル環境で実行します。スクリプトがゼロ以外の値を返すと、workload partitionは開始されません。
- **WPAR_START:** スクリプトは、workload partitionがアクティブになった後グローバル環境で実行します。アプリケーション・workload partitionsでは、スクリプトはトラッキングされたプロセスがいったん開始されると実行されます。
注記: このコード・パスは、標準 I/O ストリームをクローズまたは宛先変更して、分離処理により非同期で実行されます。内部のメッセージングは、それに応じて処理されなければならず、スクリプトは、短命のworkload partitionsがスクリプトの実行中にどこで停止または停止させられたかを説明しなければなりません。

スクリプトがゼロ以外の値を返すと、警告がログに記録されますが、それ以外に動作の変更はありません。

- **WPAR_STOP:** スクリプトは、workload partitionプロセスのすべてが終了した後で、カーネルが構成解除される前に、グローバル環境で実行されます。

注: このコード・パスは、標準 I/O ストリームをクローズして、分離処理により実行されます。内部メッセージングはそれに応じて処理されなければなりません。

スクリプトがゼロ以外の値を返すと、警告がログに記録されますが、それ以外に動作の変更はありません。

-v

冗長モードで実行するコマンドを指定します。

-x

WPAR 間セマフォおよび共用メモリー・セグメントへのアクセスを許可します。

-w

仕様ファイルを書き込みます。-o フラグとともに使用すると、-w フラグにより wparexec コマンドは workload partitionを実際に作成しないで、新規の仕様ファイルを書き込んだ後で終了します。

項目	説明
[-] [var=value] /path/to/command [arg ...]	<p>アプリケーション (トラッキングされたプロセス) がworkload partition 内で実行されるよう、必要な環境変数設定および引数とともに指定します。</p> <p>コマンドが仕様ファイルを作成している (-w フラグを使用) だけでない場合、このコマンド・ライン構文または仕様ファイル中の general.application 属性のいずれかが、このコマンドを必要とします。それが開始されたとき、提供されるコマンド・ラインは必ず、workload partition内でシェル拡張されたものになっています。コマンド・ラインを使用するとき、シェルのメタキャラクターは早期拡張を防ぐため適切に拡張する必要があります。</p> <p>特殊な二重マイナスのセパレーター (-) は、後続のすべてのコマンド・ライン引数がトラッキングされたプロセスにより成り立つことを示すために使用されます。例えば、このセパレーターを使用して、-N フラグに対する属性とトラッキングされたプロセスに対する環境変数の割り当てとの間のあいまいさを取り除きます。</p> <p>workload partitionにつき 1 つだけのトラッキングされたプロセスがサポートされますが、このアプリケーションは他のプロセスを作成することがあります。workload partitionは、ここにあるプロセスがすべて終了したときに、自動的に停止して除去されます。stopwpar コマンドによって、workload partitionの停止と除去が早まることがあります。</p>

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. ベンチマーク・プログラムを実行するアプリケーション・workload partitionを作成するには、次のように入力します。

```
wparexec -n tpcc -N address=192.168.0.51 /u/tpcc/benchmark -f /tmp/logfile
```

注: -f フラグは /u/tpcc/benchmark ファイルに渡され、wparexec コマンドによってフラグとしては処理されません。

2. 実在の仕様ファイルにもとづいてworkload partitionを作成するには、次のように入力します。

```
wparexec -f /tmp/wparexec1.spec
```

3. wparexec コマンドで提供されるデフォルトで最小限の PATH 変数をオーバーライドするには、次のように入力します。

```
wparexec PATH=/usr/opt/bin:/usr/bin:/usr/sbin /home/joe/runapp
```

ファイル

項目	説明
/usr/samples/wpars/sample.spec	注釈付きのworkload partition仕様ファイル。

関連情報:

RBAC

Trusted AIX

startwpar コマンド

route コマンド

stopwpar コマンド

wpar_reg_script コマンド

目的

このコマンドを使用すると、ユーザーはシステム WPAR Live Application Mobility のさまざまなフェーズ中に実行されるスクリプトを登録することができます。現在は、WPAR の再始動フェーズのみがサポートされています。

構文

```
/usr/lib/wpars/wpar_reg_script [ -r | -u ] [ -p phase ] [ -s /path/to/script ]
```

説明

wpar_reg_script コマンドを使用すると、ユーザーは WPAR Live Application Mobility の指定されたフェーズ中に実行されるスクリプトを登録することができます。

スクリプトを登録すると、**CuWscr** と呼ばれる新規 ODM クラスに、スクリプトへの絶対パスと引数を含むエントリーが配置されます。

このスクリプトをカーネル・エクステンションおよびモビリティとともに使用すると、他のプロセスが開始される前に、宛先ノードにカーネル・エクステンションをロードすることができます。

制約事項

wpar_reg_script コマンドは、WPAR 内からのみ実行できます。

実行時には、(専用にマウントされたファイルシステムではなく、) WPAR の構成の一部であるファイルシステムのみ使用可能です。これらのファイルシステムは、読み取り専用でのみ使用可能です。

このスクリプトが終了すると、プロセスを実行したままにしておくことはできません。

モビリティで使用するためにこのスクリプトを使用してカーネル・エクステンションをロードすることは、**SYS_SINGLELOAD** に制限されています。

フラグ

項目	説明
-r	スクリプトとフェーズを含めてエントリーを CuWscr クラスに登録します。
-u	CuWscr クラスからエントリーを登録抹消します。
-p	コマンドが実行されるフェーズ。値 1 は、WPAR の再始動に対応します。今後さらに追加される可能性があります。値が指定されていない場合は、デフォルト値の 1 が使用されます。
-s	実行されるスクリプトへのパスとその引数。コマンド・ライン・パラメーターを含めるために、コマンド全体を "" で囲んでください。

例

1. 実行するスクリプト **/usr/sbin/foo/** を登録するには、以下のように指定します。

```
/usr/lib/wpars/wpar_reg_script -r -s /usr/sbin/foo
```

2. スクリプト **/usr/sbin/foo/** を引数とともに登録するには、以下のように指定します。

```
/usr/lib/wpars/wpar_reg_script -r -s "/usr/sbin/foo args"
```

3. スクリプト **/usr/sbin/foo/** foo のすべてのインスタンスを登録抹消するには、以下のように指定します。

```
/usr/lib/wpars/wpar_reg_script -u -s /usr/sbin/foo
```

- フェーズ 1 の間に実行されるすべてのスクリプトを登録抹消するには、以下のように指定します。

```
/usr/lib/wpars/wpar_reg_script -u -p 1
```

wparprnterr コマンド

目的

ワークロード・パーティション (WPAR) に固有のエラー・メッセージを表示します。

構文

グローバル

```
wparprnterr WparName
```

説明

wparprnterr コマンドは、**wparerr** コマンド、**wpar_log_err** サブルーチン、および **kwpar_err** サブルーチンを使用して、WPAR 用にログに記録されたすべてのエラー・メッセージを標準出力上に出力します。

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。許可と特権の詳細については、「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

WPAR *mywpar* のメッセージを表示するには、次のように入力します。

```
wparprnterr mywpar
```

関連情報:

kwpar_err コマンド

lssecattr コマンド

wpar_log_err コマンド

wpar_print_err コマンド

write コマンド

目的

通信回線をオープンして、リアルタイムで、システム上の他のユーザーにメッセージを送信します。

構文

ホスト上でユーザーからの応答を待っているすべてのメッセージを照会し、そのメッセージにハンドルを付けて表示するには、次のようにタイプします。

```
write -q [ -n Host ]
```

ユーティリティーまたはシェル・スクリプトによって送信されたメッセージに応答したり、指定したハンドルに関連したメッセージを再表示するには、次のようにタイプします。

```
write -hHandle, { ok | cancel | query } [ -n Host ]
```

ユーザー (オプションで他のホスト上) または特定のデバイスにメッセージを送信するには、次のようにタイプします。

```
write [ -r ] { [ -n Host ] User | User@Host } [ Line ]
```

説明

write コマンドを使用すると、リアルタイムでメッセージをシステム上で送信できます。このコマンドによって、もう一方のログイン・ユーザーと会話式に通信することができます。各ユーザーは、他のワークステーションから短いメッセージを交互に送受信します。長いメッセージの場合は、まずメッセージ全体をファイルに入れてから、そのファイルを **write** コマンドの入力データとしてリダイレクトすると送信できます。

メッセージを受信する側のユーザー (*User* パラメーターで指定) は、ログイン済みで、メッセージ許可を拒否している状態であってはなりません。ログインしていないユーザーにメッセージを送信すると、メッセージ「user not logged in」が表示されます。**mesg** コマンドを **no** に設定してメッセージ許可を拒否しているユーザーにメッセージを送信すると、メッセージ「write: permission denied」が表示されます。

write コマンドを発行すると、下記のメッセージが、ピープ音 (ASCII BEL 文字) とともにメッセージ受信者またはターゲットに即座に送信されます。

```
Message from SenderID on SenderHostname (tty) [Date] ...
```

正常に接続すると、**write** コマンドは 2 つの ASCII BEL 文字を両方のワークステーションに送信します。ピープ音によって、送信側ユーザーにはメッセージが開始できたことを、受信側ユーザーにはメッセージを送信中であることを警告します。

Enter キーを押すごとに 1 行送信されます。送信側から受信側への通信リンクがオープン状態のままであれば、**Ctrl-D** キー・シーケンスで送信リンクを終了するまで送信は続きます。その後、テキスト終結文字 (<EOT>) が宛先ワークステーションに送信されると、**write** コマンド・モードは終了します。

受信側ユーザーまたはターゲット・ユーザーは、**write** コマンドを送り返すことで、送信元ユーザーに応答することができます。これによって、受信側から送信側へ戻す通信回線がオープンされ、返事としてのメッセージ応答が可能になります。このタイプのメッセージのやり取りでは、以下のような規則が役に立ちます。すなわち、最初に送信先ユーザーに書き出す際、テキストを送信する前に応答を待ちます。メッセージの終わりには **o** (メッセージ終了) のようなシグナルを付けて相手側ユーザーが応答するよう警告します。会話を終了するときには、**oo** (メッセージ終了および送信終了) を使用します。

行の先頭に文字 **!**(感嘆符) が付いている場合には、**write** コマンドはシェルを呼び出して残りの行をコマンドとして実行します。

複数のワークステーションにログインしているユーザーまたは複数のプロセスを多重使用しているユーザーに書き込むときには、**write** コマンドは **/etc/utmp** ファイル内に見つかった最初のログイン・インスタンスをメッセージ送達地点 (通常はログイン・シェルまたはコンソール・シェル) として使用し、下記のメッセージを表示します。

```
UserID is logged on more than one place.  
You are connected to "Workstation".  
Other locations are:  
Workstation
```

このメッセージを受け取ったら、最初のログインのロケーション以外の場所にメッセージを送信したい場合は、そのロケーションの *Line* (例えば、tty00) を指定して、別のロケーションで送信側ユーザーに接触することができます。

他のユーザーへの書き込み許可は、送信側ユーザーによって **mesg** コマンドで認可を与えられるか、拒否されるかで決まります。一部のコマンドは、出力による干渉を防止するために、実行中はメッセージ許可を拒否します。root ユーザー権限を持つユーザーは、ワークステーションのメッセージ許可に関係なく、どのワークステーションにも書き込みができます。

write コマンドを使用すると、他のホストのユーザーと会話できます。 **-nHostName** フラグまたは **User@Host** パラメーターを使用すると、リモート・ホスト上のユーザーを識別できます。リモート・ホスト上のユーザーに書き込むには、**writesrv** デーモンを、現行ホスト上とリモート・ホスト上の両方で実行中でなければなりません。

write コマンドは、**qdaemon** デーモンで使用して他のホストのユーザーにメッセージを送信し、応答を待つこともできます。有効な応答は下記の 3 つのみです。

項目	説明
ok	元の書き込みは状況 0 で終了します。
cancel	元の書き込みは状況 1 で終了します。
query	所定のハンドルに対応するメッセージが表示されます。

パラメーター

項目	説明
<i>User</i>	メッセージ・テキストを受信するユーザーのユーザー ID を指定します。
<i>User@Host</i>	メッセージ・テキストを受信するユーザーのユーザー ID およびリモート・ホストを指定します。
<i>Line</i>	別のロケーション (例えば、tty00) で送信先ユーザーと交信します。

フラグ

項目	説明
-h Handle,Reply	応答オプションのある write を使用して、ユーティリティまたはシェル・スクリプトによって送信されたメッセージに応答します。 Handle 変数に使用する値は、内部で生成され、元のメッセージのテキスト内のユーザーに与えられます。応答には、 ok 、 cancel または query が使用できます。
-nHost	リモート・ホストを指定します。 Host 変数にはニックネームまたは IP アドレスを指定できます。
-q	ホスト上でユーザーからの応答を待っているすべてのメッセージを照会し、そのメッセージにハンドルを付けて表示します。
-r	メッセージ・ハンドルを生成してメッセージのヘッダーに入れ、そのメッセージを送信して応答を待ちます。このフラグは、オペレーター・メッセージ用に qdaemon デーモンで使用され、シェル・スクリプトに入れることもできます。対話型会話には使用されません。終了状況 0 は応答が ok であったことを示し、終了状況 1 は応答が cancel であったことを示し、終了状況 2 はユーザーに接触できなかったことを示します。

要件:

- すべてのフラグを機能させるためには、ターゲット・ホスト上で **writesrv** デーモンが実行中でなければなりません。 **-n** フラグも **@Host** も使用せず、 **-h**、 **-q**、または **-r** を使用している場合は、ホスト上で **writesrv** デーモンを実行中でなければなりません。
- TCP/IP はマシンにインストールされていないが *HostName* が設定されている場合、 **-h**、 **-q**、または **-r** フラグの付いた **write** コマンドを使用してローカル・ホスト上のユーザーと会話するには、 **/etc/hosts** ファイルの **loopback** エントリーの最後に自分のホスト名を追加する必要があります。元のエントリーには下記のように記述されています。

```
127.0.0.1 loopback LocalHostName
```

新しいエントリーには下記のように記述します。

```
127.0.0.1 loopback LocalHostName HostName
```

終了状況

このコマンドは、下記の終了値を戻します。

項目	説明
0	正常終了。
>0	指定されたユーザーはログインしていないか、許可を拒否しました。

例

1. ログインしているユーザーにメッセージを書き込むには、以下のように入力します。

```
write june
```

Enter キーを押し、下記のように入力します。

```
I need to see you! Meet me in the computer room at 12:30.
```

次に、Ctrl-D キー・シーケンスを押して **write** コマンド・モードを終了します。

ユーザー ID が **karen** で、ワークステーション **tty3** を使用中であれば、 **june** のワークステーションは下記のように表示します。

```
Message from karen on trek tty3 Aug 17 11:55:24 ...
I need to see you! Meet me in the computer room at 12:30.
<EOT>
```

2. 会話をするには、以下のように入力します。

```
write june
```

Enter キーを押し、下記のように入力します。

```
Meet me in the computer room at 12:30.
o
```

これによって会話が開始します。2 行目の先頭の文字 **o** はメッセージが終了したことを示します。これは、June に応答を待っていることを知らせます。継続したい場合は、Ctrl-D を押さないでください。

June は以下のように入力して応答します。

```
write karen
```

Enter キーを押し、下記のように入力します。

```
I'm running tests at 12:30. Can we meet at 3?
o
```

ここで下記のように応答できます。

```
OK--the computer room at 3.  
oo
```

oo の意味はメッセージ終了および送信終了 で、他に伝えることがないことを June に伝えます。June も終了 (oo) したら、Ctrl-D を押して会話を終了します。

- 作成したメッセージをあるユーザーに送信するには、下記のように入力します。

```
write june < message.text
```

このコマンドは、ファイル **message.text** の内容を june のワークステーションに書き込みます。

- あるワークステーションを使用中のユーザーに書き込むには、下記のように入力します。

```
write -n console
```

Enter キーを押し、下記のように入力します。

```
The printer in building 998 has jammed.  
Please send help.
```

次に、Ctrl-D キー・シーケンスを押します。

このコマンドは、ワークステーション /dev/console にログインしているユーザーにメッセージを書き込みます。

- ホスト partya のユーザー spuds にメッセージを送信するには、下記のように入力します。

```
write -n partya spuds
```

Enter キーを押し、下記のように入力します。

```
Your new tape has just arrived,  
come see me to pick it up.  
Thanks!
```

次に、Ctrl-D キー・シーケンスを押します。

または、以下のように入力します。

```
write spuds@party
```

Enter キーを押し、下記のように入力します。

```
Your new tape has just arrived,  
come see me to pick it up.  
Thanks!
```

次に、Ctrl-D キー・シーケンスを押します。

- ここで、**qdaemon** デーモンによって送信されたメッセージの例を示します。

```
Message from mary on trek (tty10) Aug 17 10:03:34 ...  
Use "write -h 6398492,reply" to reply  
Please insert tape number 5 into rmt0.  
<EOT>
```

肯定的に応答するには、下記のように入力します。

```
write -h 6398492,ok
```

次に、Ctrl-D キー・シーケンスを押します。

否定的に応答するには、下記のように入力します。

```
write -h 6398492,cancel
```

次に、Ctrl-D キー・シーケンスを押します。

-h フラグを使用すると、ホスト名またはユーザー ID を与える必要がなくなります。この情報はハンドルにより追跡されます。

ファイル

項目	説明
<code>/etc/hosts</code>	TCP/IP ホスト情報が入っています。
<code>/etc/utmp</code>	who 、 write 、および login コマンドに関するユーザーおよびアカウント情報が入っています。

関連情報:

mesg コマンド

『writersrv デーモン』

writersrv デーモン

目的

ユーザーがリモート・システムとの間でメッセージを送受信できるようにします。

構文

writersrv

説明

writersrv デーモンを使用すると、**write** コマンドでリモート・システム上のユーザーにメッセージを送信し、リモート・システムのユーザーから応答を受信できます。

writersrv ユーティリティーは、**write** コマンドからの着信要求を受け取り、その要求を処理するサーバー・プロセスを作成します。このサーバー・プロセスは、クライアント・プロセス (**write**) と通信し、要求されたサービスを提供します。

これらのサービスを実行するには、**writersrv** デーモンは、`/etc/services` ファイル内で定義されたポートに接続されたソケットを作成します。サービスに関する要求はすべてこのソケットにメッセージとして送信されます。

注: **writersrv** デーモンが異常終了 (システム破壊、電源障害、または **kill -9** コマンドなど) した場合は、`/var/spool/writersrv` ディレクトリーを手作業で空にして、**writersrv** デーモンが残っていたファイルをすべて除去する必要があります。

例

1. `/etc/rc` スクリプトから **writersrv** デーモンを始動するには、下記のように入力します。

```
/usr/sbin/writersrv
```

writersrv デーモンは `/etc/rc` スクリプトから始動されます。これは、通常のデーモンの始動方法です。

2. システム・リソース・コントローラー (SRC) を使用して **writersrv** デーモンを始動するには、下記のように入力します。

```
startsrc -s writersrv &
```

writesrv デーモンは SRC を使用して始動されます。

ファイル

項目	説明
<code>/etc/services</code>	ネットワーク・サービス・ディレクトリーが入っています。

関連資料:

134 ページの『**write** コマンド』

関連情報:

kill コマンド

wtmpfix コマンド

目的

日付とタイム・スタンプの不整合を訂正することによって、接続時間アカウントリング・レコードを操作します。

構文

`/usr/sbin/acct/wtmpfix [File ...]`

説明

wtmpfix コマンドは、**runacct** プロシージャーによって呼び出され、**wtmp** フォーマットのレコードが入っている標準入力、つまりファイル を調べて、**acctcon1** コマンドまたは **acctcon2** コマンドの失敗の原因となる可能性がある問題を訂正します。**wtmpfix** コマンドは、日付とタイム・スタンプの不整合を訂正し、訂正後のレコードを標準出力に書き出します。**acctcon1** コマンドの実行時に、日付とタイム・スタンプが不整合である場合、**acctcon1** コマンドはエラーを生成し、処理を中止します。

また **wtmpfix** コマンドは、名前フィールドの妥当性を検査し、英数字文字、\$ (ドル記号)、またはスペースだけで構成されていることを確認します。名前が無効である場合には、**wtmpfix** コマンドはログイン名を **INVALID** に変更し、標準エラーに診断メッセージを書き出します。このように、**wtmpfix** コマンドにより、**acctcon2** コマンドが失敗する可能性は少なくなります。

(システム始動時に、または **date** コマンドにより) 日付が設定されるたびに、一組の日付変更レコードを `/var/adm/wtmp` ファイルに書き出します。最初のレコードは古い日付のレコードで、*old time* 文字列によって示されます。*old time* 文字列は直線フィールドに入れられ、**OLD_TIME** フラグはタイプ・フィールドに入れられます。2 番目のレコードは、新しい日付のレコードで、文字列 *new time* によって示されます。文字列 *new time* は、直線フィールドに入れられ、**NEW_TIME** フラグはタイプ・フィールドに入れられます。**wtmpfix** コマンドは、これらのレコードを使用して、ファイル内のすべての日付とタイム・スタンプを同期化します。

フラグ

なし。

パラメーター

項目	説明
ファイル	wtmp フォーマットのレコードが入っていることを検査したいファイルを指定します。
ル	

セキュリティ

アクセス制御: **adm** グループのメンバーだけが、これらのコマンドに実行 (x) アクセスできるようにしてください。

例

1. **/var/adm/wtmp** で日付とタイム・スタンプの不整合を修正して、**dummy.file** に修正したレコードを書き出すには、以下のように入力します。

```
/usr/sbin/acct/wtmpfix /var/adm/wtmp > /tmp/dummy.file
```

ファイル

項目	説明
/usr/sbin/acct/wtmpfix	wtmpfix コマンドが入っています。
/var/adm/wtmp	古い日付と新しい日付を持つ日付変更レコードが入っています。
/usr/include/utmp.h	理由、日付、時刻を持つヒストリー・レコードが入っています。

関連情報:

acctcon1 or acctcon2 コマンド

システム・アカウントティング

wump コマンド

目的

wumpus 狩りゲームを開始します。

構文

wump

説明

wumpus は、トンネルでつながった、たくさんの部屋を持つ洞くつに住んでいる生き物です。このゲームでは、wumpus に食べられたり底なし穴に落ちないように注意しながら、矢で wumpus を射るために部屋を動き回ります。あなたをつかまえて、ランダムに選択された部屋へ落とすスーパー・バットもいます。部屋を動き回って矢を射る場合、このゲームでは、適切な質問のプロンプトが出され、あなたの命令に従います。例えば、次のように入力します。

```
You are in room 14.
I feel a draft.
There are tunnels to 1 13 18.
Move or shoot? (m-s) m
Which room? 1
You are in room 1.
I feel a draft.
There are tunnels to 14 17 18.
Move or shoot? (m-s) m
```

```
Which room? 17
You are in room 17.
You fell into a pit!
Another game? (y-n)
```

上記の例では、ルーム 14 から開始します。コンピューターは、「I feel a draft.」を表示します。これは、穴が近くにあるというヒントです。ルーム 1 への移動を選択します。再び穴の警告が出されます。ルーム 17 への移動を選択しますが、ここで穴に落ちて死んでしまいます。

ゲームの始めに、「Instructions?(y-n)」というプロンプトが表示されます。y を選択すると、警告の説明、移動方法、および矢の発射方法が表示されます。

次の場合、ゲームが終了して「Another game? (y-n)」というプロンプトが表示されます。

- wumpus を殺したとき
- wumpus に食べられたとき
- 底なし穴に落ちたとき
- 矢を使い果たしたとき

割り込みキー・シーケンス (Ctrl-C) を押すと、ゲームをいつでも終了することができます。

ファイル

項目	説明
<code>/usr/games</code>	システムのゲームの位置。

X

以下の AIX コマンドは、文字 *x* で始まります。

X コマンド

目的

X サーバーを始動させます。

構文

```
X [ -a Number ] [ -auth FileName ] [ -bc | +bc ] [ -bp Color ] [ -broadcast ] [ -bs |  
-nobs ] [ -c Volume ] [ -cc VisualType [:Display] ] [ -class DisplayClass ] [ -co File ] [  
-cookie XDMAAuthenticationBit ] [ -D File ] [ -d Depth [:Display] ] -displayID DisplayID ] [  
-damage | +damage ] [ -evie | +evie ] [ -f Number ] [ -fc Font ] [ -fixes | +fixes ] [ -fn Font ]  
[ -fp Font ] [ -help ] [ -I ] [ -indirect HostName ] [ -layer # [:Display] ] [ -logo | nologo ] [  
-n :Number ] [ -once ] [ -P RowColumn Display } ] [ -pbuffer level [:display name | :display  
number] ] [ -p Number ] [ -port PortNumber ] [ -query HostName ] [ -r | r ] [ +render |  
-render ] [ -s Number ] [ -secIP [PermissionCode] ] [ -secLocal [PermissionCode] ] [ -secSMT  
[PermissionCode] ] [ -stereo [:Display] ] [ -su ] [ -T ] [ -t Number ] [ -to Number ] [ -v ] [ -vfb ] [ -wm ] [ -wp Color ] [ -wrap | [ -wrapx ] [ -wrapy ] ] [ -x ExtensionName ] [ -xkbdir Directory  
] [ -xkbmap FileName ] [ [+|-]accessx ] [ -ar1 Milliseconds ] [ -ar2 Milliseconds ] [ -sp FileName ] [  
+/- xinerama FileName ]
```

説明

X コマンドは X サーバーを始動します。X サーバーとは、ビットマップ端末上で実行されるディスプレイ・サーバーのことです。X サーバーは、ホスト・システムまたはネットワークでホスト・システムに接続されたシステム上にあるプログラムとの間で相互に入出力要求を配布します。

拡張 X Window システム セッションは Ctrl+Alt+Backspace キー・シーケンスを使用して終了します。

ディスプレイ・デバイスは、1 つでも複数でも指定できます。ディスプレイ・デバイスを指定しなければ、デフォルトですべてのディスプレイ・デバイスが対象になります。デフォルトの構成順は、アダプター・スロットの順序によって決まります。最初のスロット内のアダプターが左端の画面として初期化され、第 2 のスロット内のアダプターが右隣の画面となります。画面レイアウトを変更するには、**-P** フラグを使用します。**-P** フラグを使用すると、デバイスの行と桁がデバイス名に関連付けられます。**lsdisp** コマンドを使用すると、デバイス名を判別することができます。

2 台のディスプレイは縦または横に配列されます。以下の例は、横方向の配列を指定する **-P** フラグを示しています。

```
-P11 ppr0 -P12 ppr1
```

2 番めの **-P** フラグの右側の 2 は、2 つめのモニター表示が x 軸に沿って平行であることを示します。これにより、横方向の配列が生成されます。

```
Display          Display  
1                2
```

2 台のモニターを縦方向の配列で表示するには、**-P** フラグを以下のように使用する必要があります。

```
-P11 ppr0 -P21 ppr1
```

最初の位置にある 2 は、モニターが y 軸に沿って縦方向に配列されていることを示します。

```
Display
      1
Display
      2
```

横方向の構成では、マウスはディスプレイ 1 内で左から右へ移動し、ディスプレイ 1 とディスプレイ 2 のボーダーに達すると、カーソルはそのままディスプレイ 2 の同じ y 軸の位置に移動します。カーソルがディスプレイ 2 の端に達したときに、**-wrapx** フラグが設定されていると、カーソルはディスプレイ 1 の左端の同じ y 軸の位置に表示されます。**-wrapx** フラグを設定していなければ、マウスはディスプレイ 2 の右端で停止します。

縦方向の構成では、マウスはディスプレイ 1 内で上から下に移動し、ディスプレイ 1 とディスプレイ 2 のボーダーに達すると、カーソルはそのままディスプレイ 2 の同じ x 軸位置に移動します。カーソルがディスプレイ 2 の最下部に達したときに、**-wrapy** フラグが設定されていると、カーソルはディスプレイ 1 内の上端の同じ x 軸位置に表示されます。**-wrapy** フラグを設定していなければ、マウスはディスプレイ 2 の最下部で停止します。

そのほかに、情報およびエラー・メッセージ (例えば、拡張子をロードできないことを知らせるメッセージ) が **/tmp/xlogfile** ファイルにリストされています。このファイルは、X Server が問題を検出した場合に役立つ情報を提供することができます。このファイルは、X Server がインスタンスを生成するたびに再書き込みされます。このファイルは追加のエラー情報およびエラー以外の情報を提供しますが、X Server についての完全なエラー・ログではありません。

X Server は、始動時には、デフォルトのカラー・クラスを使用してる状態で立ち上がります。ドライバーによっては、X Server はデフォルトで疑似カラーまたはトゥルー・カラーのクラスがデフォルトになっている場合もあります。

疑似カラー・クラスは、画面上のカラー表示にカラー・マップを使用します。多くのグラフィックス・アダプターは、1 つのハードウェア・カラー・マップしかサポートしません。そのような場合、デフォルトのカラー・クラスが疑似カラーであって、アプリケーションがデフォルトのカラー・マップとは異なるカラー・マップを使用していると、画面に誤った色が表示されることもあります。フォーカスを持つウィンドウのみが、正しい色を表示します。画面上に誤った色が表示されないようにするには、X Server をトゥルー・カラー・モードで実行することをお勧めします。

-cc X-Server コマンド・フラグを使用して、トゥルー・カラー・クラスを使用した状態で X Server を始動させることができます。**/usr/lpp/X11/defaults/xserverrc** ファイルを以下の例に示すように変更すれば、これが可能になります。

/usr/lpp/X11/defaults/xserverrc ファイルを、root ユーザーとして編集します。 **EXTENSIONS=""** 変数を、以下の例に示すように更新します。

```
#-----
# Start the X server in True Color mode
#-----
EXTENSIONS="$EXTENSIONS -cc 4"
```

CDE からログアウトしてリセットをクリックすることにより、X Server をリスタートします。

注: **xdpinfo** コマンドを使用すれば、デフォルトのカラー・クラスを確認できます。

フラグ

項目

-a *Number*

説明

マウスの移動の加速率を指定します。例えば、5 という値を指定すると、カーソルは、マウスの 5 倍の速度で移動します。デフォルトは 4 ピクセルです。この値には、0 より大きい正の数を指定しなければなりません。

-auth *FileName*

MIT (マサチューセッツ工科大学) magic cookie を読み取るファイルに *X* を指定します。

-bc

拡張 X Window システム バージョン 1.1 への下位互換性をオフにします。

+bc

拡張 X Window システム バージョン 1.1 への下位互換性をオンにします。これはデフォルトです。

-bp *Color*

ディスプレイのブラック・ピクセル・カラーを指定します。デフォルトは、ディスプレイによって異なります。

-bs

すべての画面上でバッキング・ストア機能のサポートを使用できるようにします。デフォルトでは、バッキング・ストア機能のサポートは使用不可になっています。

-c *Volume*

キー・クリックのボリュームを指定します。

-cc *VisualType* [*:Display*]

ディスプレイ名で指定した画面のルート・ウィンドウに使用する表示タイプを指定します。すべての表示タイプが、すべてのアダプター上のすべての奥行きで使用できるわけではありません。:
Display パラメーターは任意指定ですが、マルチヘッド・オプションを使用する場合に便利です。:
Display パラメーターは、**lsdisp** コマンドで表示されるディスプレイ名です。ディスプレイ番号もディスプレイ名も指定しない場合、すべての画面に対して、指定した表示タイプが選択されます。

デフォルト表示に対し表示タイプを指定するには **-cc** フラグを、奥行きを指定するには **-d** フラグをそれぞれ使用します。

VisualType パラメーターに対する値は、以下の文字列か数値で指定します。

String Numeric equivalent

StaticGray 0

GrayScale 1

StaticColor 2

PseudoColor 3

TrueColor 4

DirectColor 5

-co *File*

赤、緑、青 (RGB) カラー・データベースの名前を設定します。これは、カラー・データベースのデフォルトのフラグです。

-D *File*

カラー定義データベース・ファイルの絶対パス名を指定します。

-d *Depth* [*:Display*]

デフォルトは **/usr/lib/X11/rgb** です。

ディスプレイ名で指定した画面のルート奥行きを指定します。すべての表示タイプを、すべてのアダプター上のすべての奥行きで使用できるわけではありません。

:
Display パラメーターは任意指定ですが、マルチヘッド・オプションを使用する場合に便利です。このパラメーターは、**-P** フラグで渡される値に対応しなければなりません。:
Display パラメーターは、**lsdisp** コマンドで表示されるディスプレイ名です。:
Display パラメーターを指定しない場合、マルチヘッド・オプション (**-P** フラグを参照) で選択したすべてのディスプレイに対して、指定した奥行きが選択されます。

-damage

X Damage 拡張機能を使用不可にします。

+damage

X Damage 拡張機能を使用可能にします。

-evie

X Event Interception 拡張機能を使用不可にします。

項目	説明										
+evie	X Event Interception 拡張機能を使用可能にします。										
-f <i>Number</i>	ビーブ音のボリュームを指定します。デフォルトは、-1 (中) です。以下の値を指定できます。										
	<table border="0"> <tr> <td>値</td> <td>設定</td> </tr> <tr> <td>0</td> <td>オフ</td> </tr> <tr> <td>1-33</td> <td>小</td> </tr> <tr> <td>-1 または 34-66</td> <td>中</td> </tr> <tr> <td>67-100</td> <td>高</td> </tr> </table>	値	設定	0	オフ	1-33	小	-1 または 34-66	中	67-100	高
値	設定										
0	オフ										
1-33	小										
-1 または 34-66	中										
67-100	高										
-fc <i>Font</i>	カーソル・グリフおよびカーソル・マスク用のカーソル・フォントを指定します。デフォルトは、オペレーティング・システムとディスプレイによって異なります。										
-fixes	X Fixes 拡張機能を使用不可にします。										
+fixes	X Fixes 拡張機能を使用可能にします。										
-fn <i>Font</i>	デフォルトのテキスト・フォントを指定します。デフォルトは、オペレーティング・システムとディスプレイによって異なります。										
-fp <i>Font</i>	フォント・パスを指定します。										
-I	コマンド・ライン上の以降の引数をすべて無効にします (i の大文字)。										
-help	使用方法メッセージを出力します。										
-layer <i>#[:Display]</i>	デフォルトの表示が # 層にあることを指定します。 <i>:Display</i> パラメーターは、 lsdisp コマンドで表示されるディスプレイ名です。オーバーレイを持たない、または 8 ビットより小さなオーバーレイを持つアダプターに対しては、このフラグを指定しても無効になります。サポートする層より大きな番号の # をこのフラグに指定すると、スクリーンのデフォルトの層にあるデフォルトの表示を指定したことになります (つまり、 -layer フラグを指定しないのと同じ)。										
-logo	スクリーンセーバーへの X Window System のロゴ表示をオンにします。現在、クライアントがこの状態を変更する方法はありません。										
-n <i>:Number</i>	接続番号を指定します。 <i>Number</i> パラメーターに有効な値は、0 から 255 までの整数です。デフォルトは、次に使用できる番号です。 <i>Number</i> パラメーターは、プログラムが特定の X サーバーと接続するときに使用されます。例えば、以下のコマンドは、										
	X -n :18										
	アクティブ化された X サーバーへの通信が、 unix:18 か Hostname:18 によって発生することを指定します。										
-nobs	すべての画面上でバッキング・ストア機能のサポートを使用不可にします。これはデフォルトです。										
nologo	スクリーンセーバーへの X Window System のロゴ表示をオフにします。現在、クライアントがこの状態を変更する方法はありません。										
-once	最初のセッションが終了した後、サーバーが終了することを指定します。通常、サーバーは自動的にセッションを開始します。										
-P <i>RowColumn Display</i>	マルチヘッド構成の中でディスプレイの物理配置を指定します。 <i>Row</i> パラメーターは、ディスプレイの行数を示します。 <i>Column</i> パラメーターは、ディスプレイの桁数を示します。										
	<i>Display</i> パラメーターは、 lsdisp コマンドの出力の最初の桁で表示されるディスプレイのデバイス名です。コマンド・ラインに出現する最初の -P <i>RowColumn Display</i> は画面 0 を X サーバーに示し、2 番目は画面 1 を示し、というようになります。										
	-P フラグは、マルチヘッド・サポートとともに使用するフラグです。										

項目

-pbuffer *level* [**:display** *name* | **:display** *number*]

-p *Number*

-r

r

+render

-render

-s*Number*

-secIP [*PermissionCode*]

-secLocal [*PermissionCode*]

-secSMT [*PermissionCode*]

説明

:display により指定された画面の、**pbuffer** メモリ割り当てレベルを指定します。このフラグは、GLX 拡張子と結合して使用した場合にのみ有効です。

level パラメーターは、フレーム・バッファ・メモリの相対的な量を **pbuffer** 用に予約するよう指示します。指定値は、0 から 2 までの範囲でなければなりません。0 の値は、**pbuffer** 用にメモリを予約しないよう指示します。1 の値は、少量のメモリを予約するよう指示します。2 の値は、多量のメモリを予約するよう指示します。すべてのアダプターが **pbuffer** をサポートするわけではありません。サポートしているアダプターでも、すべての画面構成が **pbuffer** をサポートするわけではありません。**pbuffer** 用に予約されるフレーム・バッファ・メモリの実際の量は、デバイスによって決まり、画面解像度やデフォルトのピクセルの密度のような、他の要素の影響を受ける場合があります。

:display パラメーターは任意指定ですが、マルチヘッド・オプションを使用する場合に便利です。**:display** パラメーターは、**lsdisp** コマンドで表示されるディスプレイ名です。ディスプレイの *number* も *name* も指定しない場合、指定の **pbuffer** 幅がすべての画面に対して選択されます。

X Window System のロゴの位置が変わる間隔を分単位で指定します。このフラグは、**-s** (スクリーンセーバー・タイムアウト) フラグと併用して、スクリーンのブランキングを制御します。オートリピートを使用不可にします。デフォルトでは、オートリピートは使用可能です。

オートリピートを使用可能にします。

X レンダー拡張機能を使用可能にします。デフォルトでは、X レンダー拡張機能は使用不可です。

注: X レンダー拡張機能は、GXT135P、GXT145、GXT4500P、および GXT6500P のグラフィックス・アダプターを装備した X サーバーのみで活動化できます。システム上で使用可能なアダプターを確認するには、**lsdisp** コマンドを実行してください。

+render フラグと共に **-vfb** フラグを指定して、仮想フレーム・バッファを使用します。

X レンダー拡張機能を使用不可にします。

スクリーンがブランキングするまで待機する時間を分単位で指定します。デフォルトは 10 分です。この値に 0 を設定した場合、スクリーンセーバーは使用不可になります。

インターネット・ソケットでローカル・アクセス制御を設定します。*PermissionCode* は 3 桁の 8 進数字であり、読み取り、書き込み、および実行のビットを設定することができます。セキュリティー・フラグの後に *PermissionCode* が指定されていないと、そのソケットに対して、アクセス権はデフォルトで 0 になります。UNIX ソケットでアクセス制御を設定します。*PermissionCode* は 3 桁の 8 進数字であり、読み取り、書き込み、および実行のビットを設定することができます。セキュリティー・フラグの後に *PermissionCode* が指定されていないと、そのソケットに対して、アクセス権はデフォルトで 0 になります。

共用メモリ・トランスポート・ソケットでアクセス制御を設定します。*PermissionCode* は 3 桁の 8 進数字であり、読み取り、書き込み、および実行のビットを設定することができます。セキュリティー・フラグの後に *PermissionCode* が指定されていないと、そのソケットに対して、アクセス権はデフォルトで 0 になります。

項目

-stereo [:*Display*]

説明

Display によって指定された画面用の、最適のステレオ・サポートのためのグラフィックス・アダプターを構成します。

サポートされる画面は、ステレオを最良にサポートするようにアダプターを構成します。これにより、テクスチャー・メモリーのような他のリソースが減少することがあります。影響を受ける実際のメモリーの量はデバイスによって決まり、画面解像度やデフォルトのピクセルの密度のような、他の要素の影響を受ける場合があります。

Display パラメーターは任意指定ですが、マルチヘッド・オプションを使用する場合に便利です。 *Display* パラメーターは、**lsdisp** コマンドで表示されるディスプレイ名です。ディスプレイ番号もディスプレイ名も指定しない場合、**-stereo** フラグがサポートされるすべてのスクリーンに対して適用されます。

-su

-T

サポートされない画面は、**-stereo** フラグを無視します。

すべての画面上でサポート付き保存を使用不可にします。

Ctrl+Alt+Backspace キー・シーケンスを使用不可にします。このキー・シーケンスは、デフォルトでは、**AIXwindows** セッションと、そこからオープンされたすべてのウィンドウを終了させます。

-t *Number*

マウスしきい値を指定します。デフォルトは 2 ピクセルです。一定の時間の間にマウスが移動した距離がこのマウスのしきい値を超えたとき、加速が行われます。また、加速は、しきい値を超えた距離にしか適用されません。

-to *Number*

接続検査の経過時間を分単位で指定します。デフォルトは 60 分です。指定した値は 0 より大きな値でなければなりません。

-v

-s フラグで指定した時間後に、表示が現在のバックグラウンド・カラーに置き換えられることを指定します。デフォルトでは、**-v** フラグが使用されていないならば、**-s** フラグで指定した時間後に、ディスプレイ全体がバックグラウンド・タイルでペイントされます。

-vfb

グラフィックス・アダプターを初期化せず、仮想フレーム・バッファ (VFB) を使用して X サーバーを始動します。

-wm

すべてのウィンドウのデフォルトのパッキング・ストアに

WhenMapped 値を強制設定します。このフラグは、すべてのウィンドウにパッキング・ストアを適用するときに便利です。

-wp *Color*

ホワイト・ピクセル・ディスプレイ・カラーを指定します。デフォルトは、ディスプレイによって異なります。

項目
-wrap

説明

マウスのホット・スポットがルート・ウィンドウの上下左右いずれかのボーダーに達したときのマウスの動作を指定します。このフラグが設定されている場合、マウスのホット・スポットが一番左にあるルート・ウィンドウの左端に達したとき、マウスは自動的に、一番右にあるルート・ウィンドウの右端 (y 座標は同じ) に移動します。

また逆に、このフラグが設定されていれば、マウスのホット・スポットが一番右にあるルート・ウィンドウの右端に達したとき、マウスは自動的に、一番左にあるルート・ウィンドウの左端 (y 座標は同じ) に移動します。このフラグが設定されていなければ、どのルート・ウィンドウでもマウスは左右どちらかの端で止まります。

このフラグが設定されている場合、マウスのホット・スポットが一番上にあるルート・ウィンドウの上端に達したとき、マウスは自動的に、一番下にあるルート・ウィンドウの下端 (x 座標は同じ) に移動します。

また逆に、このフラグがセットされて、マウスのホット・スポットが一番下にあるルート・ウィンドウの下端に達したとき、マウスは自動的に、一番上にあるルート・ウィンドウの上端 (x 座標は同じ) に移動します。

-wrapx

-wrap フラグは、マルチヘッド・サポートとともに使用するフラグです。

マウスのホット・スポットがルート・ウィンドウの左右いずれかのボーダーに達したときのマウスの動作を指定します。このフラグが設定されている場合、マウスのホット・スポットが一番左にあるルート・ウィンドウの左端に達したとき、マウスは自動的に、一番右にあるルート・ウィンドウの右端 (y 座標は同じ) に移動します。また逆に、このフラグが設定されていると、マウスのホット・スポットが一番右にあるルート・ウィンドウの右端に達したとき、マウスは自動的に、一番左にあるルート・ウィンドウの左端 (y 座標は同じ) に移動します。このフラグが設定されていなければ、どのルート・ウィンドウでもマウスは左右どちらかの端で止まります。

-wrapy

-wrapx フラグは、マルチヘッド・サポートとともに使用するフラグです。

マウスのホット・スポットがルート・ウィンドウの上下いずれかのボーダーに達したときのマウスの動作を指定します。このフラグが設定されている場合、マウスのホット・スポットが一番上にあるルート・ウィンドウの上端に達したとき、マウスは自動的に、一番下にあるルート・ウィンドウの下端 (x 座標は同じ) に移動します。

また逆に、このフラグがセットされて、マウスのホット・スポットが一番下にあるルート・ウィンドウの下端に達したとき、マウスは自動的に、一番上にあるルート・ウィンドウの上端 (x 座標は同じ) に移動します。このフラグが設定されていなければ、マウスはどのルート・ウィンドウでも上下どちらかの端で止まります。

-wrapy フラグは、マルチヘッド・サポートとともに使用するフラグです。

項目	説明
-x <i>ExtensionName</i>	サーバーが初期化されるときにロードする拡張名を指定します。これは、Display PostScript Level 2 (dps) のような大規模な拡張機能には有用です。複数の拡張名を指定するときは、このフラグを複数指定することができます。
-query <i>HostName</i>	拡張 X Window システム Display Manager 制御プロトコル (XDMCP) を使用可能にし、指定されたホストに Query パケットを送信します。
-broadcast	-query フラグは、 XDMCP とともに使用するフラグです。 XDMCP を使用可能にし、 BroadcastQuery パケットをネットワークにブロードキャストします。最初に応答した Display Manager がセッション用に選択されます。
-indirect <i>HostName</i>	-broadcast フラグは、 XDMCP とともに使用するフラグです。 XDMCP を使用可能にし、 IndirectQuery パケットを指定したホストに送信します。
-port <i>PortNumber</i>	-indirect フラグは、 XDMCP とともに使用するフラグです。 XDMCP 用の代替ポート番号を指定します。このフラグは、 -query フラグ、 -broadcast フラグ、 -indirect フラグのどれよりも先に指定しなければなりません。通常、サーバーは 1 つずつセッションを始動します。このフラグを指定すると、最初のセッションが終了した後、サーバーが終了します。
-class <i>DisplayClass</i>	-port フラグは、 XDMCP とともに使用するフラグです。 XDMCP がディスプレイ固有オプションのリソース索引で使用する追加のディスプレイ修飾子に値を設定します。
-cookie <i>XDMAAuthenticationBits</i>	-class フラグは、 XDMCP とともに使用するフラグです。 XDM-AUTHENTICATION-1 を検査するときに、サーバーとマネージャーとの間で共有される秘密鍵を指定します。
-displayID <i>DisplayID</i>	-cookie フラグは、 XDMCP とともに使用するフラグです。Display Manager が -cookie フラグで指定した共有キーを配置できるディスプレイを識別するためのディスプレイ識別コードを指定します。
+/- xinerama	-displayID フラグは、 XDMCP とともに使用するフラグです。パノラマ・スクリーンまたは仮想ラージ・スクリーン (VLS) を使用可能または使用不可にします。ユーザーは、すべてのマルチヘッド環境のすべてのヘッドをラージ・スクリーンとして扱うことができます。

Xkeyboard フラグ

項目	説明
-xkbdir <i>Directory</i>	キーボード・レイアウト・ファイルの基本ディレクトリーを指定します。
-xkbmap <i>FileName</i>	始動時にロードするキーボード説明を指定します。
[+ -]accessx	AccessX キー・シーケンスを使用可能 (+) または使用不可 (-) にします。
-ar1 <i>Milliseconds</i>	オートリピートを開始するまでにキーを押し続ける時間の長さ (ミリ秒単位) を設定します。
-ar2 <i>Milliseconds</i>	オートリピートを生成するキー・ストロークとキー・ストロークの間で、経過させる時間の長さ (ミリ秒単位) を設定します。

セキュリティ拡張フラグ

項目

-sp *FileName*

説明

サーバーが *FileName* を読み取り、下記のフォーマットでセキュリティー・ポリシー・ファイルとして解釈するようになります。ファイルはサーバー始動時に読み取られ、サーバーのリセットごとに再読み取りされます。

セキュリティー・ポリシー・ファイルの構文は下記のとおりです。表記法: "*" は先行するエレメントのゼロ以上のオカレンスを意味し、"+" は 1 つ以上のオカレンスを意味します。 *foo/bar* の解釈では、/以降のテキストを無視します。これは、次のセクションの *foo* のインスタンスを区別するために使用されています。

```
policy file ::= version line other line*  
version line ::= string/v '\n'  
other line ::= comment | access rule | site policy | blank line  
comment ::= # not newline* '\n'  
blank line ::= space '\n'  
site policy ::= sitepolicy string/sp '\n'  
access rule ::= property property/ar window perms '\n'  
property ::= string  
window ::= any | root | required property  
required property ::= property/rp | property with value  
property with value ::= property/rpv = string/rv  
perms ::= [ operation | action | space ]*  
operation ::= r | w | d  
action ::= a | i | e  
string ::= dbl quoted string | single quoted string | unquoted string  
dbl quoted string ::= space " not dqoute* " space  
single quoted string ::= space ' not squote* ' space  
unquoted string ::= space not space+ space  
space ::= [ ' ' | '\t' ]*
```

文字セット:

```
not newline ::= any character except '\n'  
not dqoute ::= any character except "  
not squote ::= any character except '  
not space ::= any character except those in space
```

前述の構文に関連したセマンティクスは、下記のとおりです。

version line

ファイルのこの最初の行は、ファイル・フォーマットのバージョンを示します。サーバーは、バージョン *string/v* を認識しない場合、残りのファイルを無視します。ここで説明しているファイル・フォーマットのバージョン文字列は、*version-1* です。

version line から先は、上記の構文に一致しない行は無視されます。

comment

行は無視されます。

sitepolicy

現在の行は無視されます。これらは、XC-QUERY-SECURITY-1 許可メソッドによって使用されるサイト・ポリシーを指定するためのものです。

access rule

この行は、*property/ar* という名前の X Window 属性に影響を与える、非トラステッド・クライアントの要求にサーバーが反応する方法を指定します。このセクションの残りの部分は、*access rule* の解釈を説明します。

property/ar の特定のインスタンスに適用する *access rule* の場合、*property/ar* は、*window* によって指定される一連のウィンドウ内のウィンドウ上になければなりません。*window* が **any** の場合、この規則はすべてのウィンドウ上の *property/ar* に適用されます。*window* が **root** の場合、この規則はルート・ウィンドウ上の *property/ar* に適用されます。

window が *required property* の場合、下記が適用されます。*required property* が *property/rp* の場合、値に関係なく、この規則はウィンドウに *property/rp* もある場合に適用されます。*required property* が *property with value* の場合、*property/rpv* は、*string/rv* によって指定された値も持たなければなりません。この場合、属性は STRING のタイプで、フォーマットは 8 にする必要があり、1 つ以上の **null** で終了する文字列を含める必要があります。いずれかの文字列が *string/rv* に一致する場合、この規則が適用されます。

文字列突き合わせの定義は、単純な大/小文字を区別する文字列比較です。*string/rv* で文字 **'*'** のオカレンスは、「任意の文字列」を意味するワイルドカードです。*string/rv* には、文字列の任意の場所に複数のワイルドカードを含めることができます。例えば、**x*** は **x** で始まる文字列に一致し、***x** は **x** で終わる文字列に一致し、***x*** は **x** を含む文字列に一致し、**x*y*** は **x** で始まり、その後 **y** を含む文字列に一致します。

特定の *property/ar* に複数の *access rule* を存在させることができます。規則は、ファイル内の順序でテストされます。一致する最初の規則が使用されます。

perms 非トラステッド・クライアントが試行できる操作と、それらの操作に応じてサーバーが取るアクションを指定します。

operation

r (読み取り)、**w** (書き込み)、または **d** (削除) のいずれかにできます。下記の情報は、X プロトコル属性要求が、X コンソーシアム・サーバー・インプリメンテーションでの操作にどのようにマップするかを示しています。

GetProperty

r、または **delete = True** の場合は **r** および **d**

ChangeProperty

w

RotateProperties

r および **w**

DeleteProperty

d

ListProperties

none、非トラステッド・クライアントは常にすべての属性をリストできます。

action **a** (許可)、**i** (無視)、または **e** (エラー) のいずれかにできます。

Allow 要求を、トラステッド・クライアントが発行したもののよう実行します。

Ignore 要求をノーオペレーションとして扱います。GetProperty の場合、ignore は、その実際の値に関係なく、属性が存在する場合は空の属性値を戻すことを意味します。

Error 要求を実行せず、atom を属性名に設定して BadAtom エラーを戻すように指定します。エラーは、セキュリティー・ポリシー・ファイルにリストされていないものも含め、すべての属性のデフォルトのアクションです。

action は、それ以降、次の *action>* が検出されるまでのすべての *operation* に適用されます。したがって、*irwad* は、ignore read および write、allow delete を意味します。

GetProperty および RotateProperties は、複数の操作を実行する場合があります (**r** および **d**、または **r** および **w**)。さまざまなアクションが操作に適用される場合、最も重大なアクションが要求全体に適用されます。部分的な要求の実行はありません。重大度の順序は、allow (許可) < ignore (無視) < error (エラー) です。したがって、属性の *perms* が *ired* (ignore read, error delete) であり、非トラステッド・クライアントが、その属性に対して GetProperty を **delete = True** で試行する場合、エラーは戻されますが、属性値は戻されません。同様に、RotateProperties のいずれかの属性が、読み取りおよび書き込みの両方を許可しない場合、エラーは属性値を変更せずに戻されます。

以下はセキュリティー・ポリシー・ファイルの例です。

version-1

アプリケーション・リソースは読み取ることができますが、書き込むことはできません。

```
property RESOURCE_MANAGER    root    ar iw
property SCREEN_RESOURCES    root    ar iw
```

CUT バッファの使用を試みても無視されます。エラーを与えると apps はクラッシュし、アクセスを許可すると多すぎる情報が与えられます。

```
property CUT_BUFFER0         root    irw
property CUT_BUFFER1         root    irw
property CUT_BUFFER2         root    irw
property CUT_BUFFER3         root    irw
property CUT_BUFFER4         root    irw
property CUT_BUFFER5         root    irw
property CUT_BUFFER6         root    irw
property CUT_BUFFER7         root    irw
```

Motif を使用している場合は、これらが必要でしょう。

```
property _MOTIF_DEFAULT_BINDINGS    rootar iw
property _MOTIF_DRAG_WINDOW          root    ar iw
property _MOTIF_DRAG_TARGETS        any     ar iw
property _MOTIF_DRAG_ATOMS          any     ar iw
property _MOTIF_DRAG_ATOM_PAIRS     any     ar iw
```

下記の 2 つのルールは、非トラステッドの場合、xwininfo -tree work を行います。

```
property WM_NAME                any     ar
```

WM_CLASS の読み取りを許可、WM_NAME のウィンドウのみ。

これは必要以上に限定的かもしれませんが、
必要な属性 機能をデモンストレーションし、さらに
"トップレベルのウィンドウのみ" と表示させます。

```
property WM_CLASS                WM_NAME  ar
```

次の 3 つは xlsclients 作業を非トラステッドにします。

これらを含める前には、十分な考慮が必要です。クライアントのマシン名や
コマンドを与えると多すぎるほどの情報が得られます。

```
property WM_STATE                WM_NAME  ar
property WM_CLIENT_MACHINE       WM_NAME  ar
property WM_COMMAND              WM_NAME  ar
```

非トラステッド・クライアントに、

xstdcmap で作成した標準カラー・マップを使用させるには、下記の行を指定します。

```
property RGB_DEFAULT_MAP        root    ar
property RGB_BEST_MAP           root    ar
property RGB_RED_MAP            root    ar
property RGB_GREEN_MAP         root    ar
property RGB_BLUE_MAP           root    ar
property RGB_GRAY_MAP           root    ar
```

項目 説明

```
# 非トラステッド・クライアントに xcmsdb で作成した
# カラー・マネージメント・データベースを使用させるには、下記の行を指定します。
property XDCCC_LINEAR_RGB_CORRECTION rootar
property XDCCC_LINEAR_RGB_MATRICES rootar
property XDCCC_GRAY_SCREENWHITEPOINT rootar
property XDCCC_GRAY_CORRECTION rootar

# 変わった属性名やエラー条件の明示的な指定
property "property with spaces" 'property with "law er ed

# ウィンドウが "son" で終わる値の属性 0hBoy も持つ場合、Woo-Hoo を削除可能です。
# 読み取りおよび書き込みは、エラーを発生するでしょう。
property Woo-Hoo 0hBoy = "*son"ad
```

例

X レンダー拡張機能を使用して X サーバーを始動するには、以下のコマンドを入力してください。

```
$X -T -force :0 -vfb -d 32 +render
```

この例では、X サーバーは、レンダリング用に物理的なグラフィックス・アダプターを使用する代わりに、仮想フレーム・バッファ (VFB) を使用します。

関連情報:

lsdisp コマンド

aixterm コマンド

x_add_fs_fpe コマンド

目的

ネットワーク・フォント・サーバーをフォント・パスに追加します。

構文

```
x_add_fs_fpe Host Port Position TypeName
```

説明

x_add_fs_fpe コマンドは、フォント・サーバーがフォントにアクセスできるように、選択したネットワーク・タイプ名のフォント・パスにフォント・パス・エレメントを追加します。

項目	説明
<i>Host</i>	フォント・サーバーが存在するシステムの名前を指定します。
<i>Port</i>	フォント・サーバー・ポートの番号を指定します。この番号は /etc/services ファイルにある番号でなければならず、10 進数で指定します。
<i>Position</i>	このエレメントを挿入するフォント・パス内の位置を指定します。
<i>TypeName</i>	ネットワーク・タイプの名前を指定します。ネットワーク・タイプごとに、1 つ以上のフォント・パス・エレメントからなるフォント・パスがあります。フォント・パス・エレメントを追加するネットワーク・タイプの名前を指定します。または、すべてのネットワーク・タイプ名にフォント・パスを追加する場合は、A11 を指定します。フォント・パス・エレメントをすべての (A11) ネットワーク・タイプに追加する場合は、各フォント・パスの最後に追加されます。

セキュリティ

アクセス制御 : root ユーザーだけが、このコマンドに実行 (x) アクセスできるようにしてください。

例

フォント・サーバーを、ネットワーク・タイプ `x_st_mgr.ether` のフォント・パスの始めに追加するには、下記のように入力します。

```
x_add_fs_fpe winter 7500 1 x_st_mgr.ether
```

この例では、ホスト `winter` 上のフォント・サーバーがネットワーク・タイプ `x_st_mgr.ether` のフォント・パスの始めに追加されます。フォント・サーバー・ポートは `7500` です。

ファイル

項目	説明
<code>/usr/lpp/x_st_mgr/bin/x_add_fs_fpe</code>	<code>x_add_fs_fpe</code> コマンドが入っています。
<code>/etc/x_st_mgr/ether.cf</code>	ネットワーク・タイプ <code>x_st_mgr.ether</code> 構成ファイル (サンプル) が入っています。

関連資料:

『`x_add_nfs_fpe` コマンド』

157 ページの『`x_rm_fpe` コマンド』

x_add_nfs_fpe コマンド

目的

NFS/TFTP でアクセスするフォント・ディレクトリーをフォント・パスに追加します。

構文

```
x_add_nfs_fpe Host Directory Method Position TypeName
```

説明

`x_add_nfs_fpe` コマンドは、選択したネットワーク・タイプ名のフォント・パスにフォント・パス・エレメントを追加します。このフォント・ディレクトリーには、ネットワーク・ファイルシステム (NFS) またはトリビアル・ファイル転送プロトコル (TFTP) を使ってアクセスします。

項目	説明
<code>Host</code>	フォント・ディレクトリーにアクセスするためのシステム名を指定します。
<code>Directory</code>	フォントが入っているディレクトリーへの絶対パスを指定します。
<code>Method</code>	フォントへのアクセスに使う <code>nfs</code> または <code>tftp</code> のいずれかを指定します。
<code>Position</code>	このエレメントを挿入するフォント・パス内の位置を指定します。
<code>TypeName</code>	ネットワーク・タイプの名前を指定します。ネットワーク・タイプごとに、1 つ以上のフォント・パス・エレメントからなるフォント・パスがあります。フォント・パス・エレメントを追加するネットワーク・タイプの名前を指定します。または、すべてのネットワーク・タイプ名にフォント・パスを追加する場合は、 <code>A11</code> を指定します。フォント・パス・エレメントをすべての (A11) ネットワーク・タイプに追加すると、各フォント・パスの最後に追加されます。

セキュリティ

アクセス制御 : root ユーザーだけが、このコマンドに実行 (x) アクセスできるようにしてください。

例

/usr/lib/X11/fonts/100dpi 内のフォントを、ネットワーク・タイプ `x_st_mgr.ether` に追加するには、下記のように入力します。

```
x_add_nfs_fpe cedar /usr/lib/X11/fonts/100dpi nfs Last ¥ x_st_mgr.ether
```

この例では、フォント・パス・エレメント `/usr/lib/X11/fonts/100dpi` がネットワーク・タイプ `x_st_mgr.ether` のフォント・パスの最後に追加されます。フォント・ディレクトリーはホスト `cedar` 上にあり、このホストには NFS を使ってアクセスします。

ファイル

項目	説明
<code>/usr/lpp/x_st_mgr/bin/x_add_nfs_fpe</code>	<code>x_add_nfs_fpe</code> コマンドが入っています。
<code>/etc/x_st_mgr/ether.cf</code>	ネットワーク・タイプ <code>x_st_mgr.ether</code> 構成ファイル (サンプル) が入っています。

関連資料:

155 ページの『`x_add_fs_fpe` コマンド』

『`x_rm_fpe` コマンド』

x_rm_fpe コマンド

目的

フォント・パスからフォント・パスのエレメントを除去します。

構文

```
x_rm_fpe TypeName Position Method Host Post Directory
```

説明

`x_rm_fpe` コマンドは、選択されたネットワーク・タイプ名のフォント・パスから、フォント・パス・エレメントを除去します。

項目	説明
<i>TypeName</i>	エレメントを除去するネットワーク・タイプ名を指定します。
<i>Position</i>	フォント・パス内のエレメントの位置を指定します。
<i>Method</i>	フォント・パス・エレメントにアクセスする場合の方式を指定します。有効なオプションは以下のとおりです。ネットワーク・フォント・サーバーの場合は tcp 、初期デフォルト・フォント・パス・エレメントの場合は default 、NFS の場合は nfs 、および TFTP の場合は tftp です。
<i>Host</i>	フォント・パス・エレメント内に指定されたシステム名を指定します。デフォルトのアクセス方式を使用するエレメントの場合は、 None を指定します。
<i>Port</i>	フォント・パス・エレメント内に指定されたサーバー・ポート番号を指定します。nfs または tftp を使用するエレメントの場合は、 None を指定します。
<i>Directory</i>	フォントが入っているディレクトリーへの絶対パスを指定します。ネットワーク・フォント・サーバー・エレメントの場合は、 None 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、このコマンドに実行 (x) アクセスできるようにしてください。

例

ネットワーク・タイプ `x_st_mgr.ether` のフォント・パスからフォント・エレメント `/usr/lib/X11/fonts/100dpi` を除去するには、以下のように入力します。

```
x_rm_fpe x_st_mgr.ether 3 nfs waco None /usr/lib/X11/fonts/100dpi
```

この例では、NFS を使ってホスト `waco` 上でアクセスされるフォント・パス・エレメント `/usr/lib/X11/fonts/100dpi` が、ネットワーク・タイプ `x_st_mgr.ether` のフォント・パスの 3 番目の位置から除去されます。NFS にはポート番号が使用されないため、このパラメーターは `None` に設定されています。

ファイル

項目	説明
<code>/usr/lpp/x_st_mgr/bin/x_rm_fpe</code>	<code>x_rm_fpe</code> コマンドが入っています。
<code>/etc/x_st_mgr/ether.cf</code>	ネットワーク・タイプ <code>x_st_mgr.ether</code> 構成ファイル (サンプル) が入っています。

関連資料:

156 ページの『`x_add_nfs_fpe` コマンド』

157 ページの『`x_rm_fpe` コマンド』

xargs コマンド

目的

パラメーター・リストを作成し、コマンドを実行します。

構文

```
xargs [-p t x ] [-e [EOFString]] [-E EOFString] [-i [ReplaceString] ] [-I ReplaceString | -L Number | -n Number] [ -l [ Number]] [-s Size] [Command [Argument ... ]]
```

注: 小文字フラグとパラメーターの間にはブランク・スペースを入れないでください。

説明

生成されるコマンド・ラインの長さは、*Command* と、各文字列の `null` バイト・ターミネーターを含む、文字列として処理される各 *Argument* のバイト数の和です。`xargs` コマンドは、コマンド・ラインの長さを制限します。作成されたコマンド・ラインを実行する場合に、*Argument* と環境を組み合わせたリストの長さは、**ARG_MAX** バイトを超えることはできません。`-n` または `-s` フラグを指定しないと、デフォルトのコマンド・ラインの長さは、この制限の範囲内で **LINE_MAX** で指定した値以上となります。

フラグ

項目	説明
-e [EOFString]	<p>旧バージョンのフラグ。 -E フラグを使用してください。</p> <p><i>EOFString</i> パラメーターを論理 EOF 文字列として使用します。 -e または -E フラグを指定しなければ、論理 EOF 文字列には下線文字 () が想定されます。 <i>EOFString</i> パラメーターを指定しなければ、論理 EOF 文字列機能は使用不可になり、下線文字はリテラル文字として解釈されます。 xargs コマンドは、EOF 文字または指定された文字列に達するまで、標準入力を読み取ります。</p>
-E EOFString	<p>デフォルトの下線 () を置き換える論理 EOF 文字列を指定します。 xargs コマンドは、EOF 文字または指定された文字列に達するまで標準入力を読み取ります。</p>
-i [ReplaceString]	<p>旧バージョンのフラグ。 -I (i の大文字) フラグを使用してください。</p> <p><i>ReplaceString</i> パラメーターを指定しなければ、文字列 {} が使用されます。</p> <p>注: -I (i の大文字)、i、-L (l の大文字)、l、および -n の各フラグは相互に排他的です。最後に指定されたフラグが有効になります。</p>
-I ReplaceString	<p>(i の大文字)。標準入力の各行を <i>Command</i> パラメーターの引数として挿入 (つまり、出現する各 <i>ReplaceString</i> に対する <i>Argument</i> に挿入) します。 <i>ReplaceStrings</i> は、6 個以上の引数で使用できません。各標準入力行の先頭の空白文字は無視されます。各 <i>Argument</i> には、1 つ以上の <i>ReplaceStrings</i> を含むことができますが、255 バイト以上にはなりません。 -I フラグは、-x フラグもオンにします。</p> <p>-I (i の大文字) フラグは -L1 を意味します。したがって、一度に 1 つだけの標準入力行を引数として置換できます。置換された文字列がコマンド・パラメーターで 2 回以上表示される場合、置換された文字列のオカレンスごとに、同じ標準入力行が置換されます。</p> <p>注: -I (i の大文字)、i、-L (l の大文字)、l、および -n の各フラグは相互に排他的です。最後に指定されたフラグが有効になります。</p>
-l [Number]	<p>1 (L の小文字)。旧バージョンのフラグ。 -L フラグを使用してください。</p> <p><i>Number</i> パラメーターを指定しなければ、値 1 が使用されます。また、-l フラグは -x フラグもオンにします。</p> <p>注: -I (i の大文字)、i、-L (l の大文字)、-l、および -n の各フラグは相互に排他的です。最後に指定されたフラグが有効になります。</p>
-L Number	<p>空でないパラメーター行を指定された数だけ標準入力から読み取って、<i>Command</i> パラメーターを実行します。残りの <i>Number</i> が指定された数よりも少なければ、<i>Command</i> を最後に呼び出すときに、パラメーター行の数を減らすことができます。行の最後の文字がスペースまたはタブでない限り、行は最初の改行文字で終了します。後続スペースは、次の空でない行への継続を示します。</p> <p>標準入力から <i>Number</i> 行を読み取り、コマンド・ラインの末尾に置きます。</p> <p>注: -I (i の大文字)、i、-L (l の大文字)、-l、および -n の各フラグは相互に排他的です。最後に指定されたフラグが有効になります。</p>
-n Number	<p>できるだけ多くの標準入力引数を使用して、<i>Number</i> パラメーターで指定された範囲内で、<i>Command</i> パラメーターを実行します。</p> <p>標準入力から <i>Number</i> 引数の最大値を読み取り、コマンド・ラインの末尾に置きます。以下の場合には、xargs コマンドで使用する引数の数が減少します。</p> <ul style="list-style-type: none"> • 集計されたコマンド・ラインの長さが -s <i>Size</i> フラグで指定したバイト数を超える場合 • 最後の反復で残った引数の値が <i>Number</i> 引数で指定した値よりも小さいが、ゼロ以外の場合。 <p>注: -I (i の大文字)、i、-L (l の大文字)、-l、および -n の各フラグは相互に排他的です。最後に指定されたフラグが有効になります。</p>
-p	<p><i>Command</i> パラメーターを実行するかどうかを尋ねます。作成されたコマンド・ラインに続けて、?... (疑問符と省略符号) プロンプトを表示します。 <i>Command</i> パラメーターを実行するには、ロケールに固有の肯定応答を入力します。他の応答を入力すると、xargs コマンドはパラメーターの特定の呼び出しをスキップします。呼び出しごとにプロンプトが表示されます。 -p フラグは、-t フラグもオンにします。</p>

項目	説明
-s <i>Size</i>	作成される <i>Command</i> 行の最大合計サイズを設定します。 <i>Size</i> パラメーターには、必ず正の整数を指定してください。以下の場合には、使用される引数の数が減少します。 <ol style="list-style-type: none"> 1. 引数の合計数が、-n フラグで指定した数を超える場合。 2. 行の合計数が -L フラグまたは -l (L の小文字) フラグで指定した行数を超える場合。 3. <i>Size</i> パラメーターで指定したバイト数まで累積する前に、EOF 文字に達した場合。
-t	トレース・モードを使用可能にして、作成済みの <i>Command</i> 行を標準エラーにエコーしてから実行します。
-x	<i>Command</i> 行が -s <i>Size</i> フラグで指定されたバイト数より大きい場合に、 xargs コマンドの実行を停止します。 -I (i の大文字) フラグまたは -l (L の小文字) フラグのいずれかを指定すると、この -x フラグがオンになります。 -i 、 -I (i の大文字)、 -l (L の小文字)、 -L または -n フラグを指定しないと、 <i>Command</i> 行の合計の長さは、 -s <i>Size</i> フラグで指定した限度を超えません。

終了状況

このコマンドは、下記の終了値を戻します。

項目	説明
0	<i>Command</i> パラメーターのすべての呼び出しが、終了状況 0 を戻しました。
1-125	指定された要件を満たすコマンド・ラインをアセンブルできなかったか、 <i>Command</i> パラメーターの 1 回または数回の呼び出しで、ゼロ以外の終了状況が戻されたか、またはその他のエラーが発生しました。
126	<i>Command</i> を見つけましたが、呼び出せません。
127	<i>Command</i> が見つかりません。

指定された要件を満たすコマンド・ラインをアセンブルできない場合、コマンドの呼び出し不能、シグナルによる打ち切り、終了状況 **255** による終了のいずれかが起こります。**xargs** コマンドは、残りの入力をも処理せずに、診断メッセージを書き込んで終了します。

例

1. ファイル内で名前がリストされるファイル上でコマンドを使用するには、以下のように入力します。

```
xargs lint -a <cfiles
```

cfiles ファイルに以下のテキストが入っているとします。

```
main.c readit.c
gettoken.c
putobj.c
```

xargs コマンドは以下のコマンドを作成して実行します。

```
lint -a main.c readit.c gettoken.c putobj.c
```

cfiles ファイルに入っているファイル名が多すぎて、シェル・コマンド 1 行 (最大 **LINE_MAX**) に入りきれない場合、**xargs** コマンドは 1 行におさまるファイル名を使用して **lint** コマンドを実行します。次に、残りのファイル名を使用して、別の **lint** コマンドを作成して実行します。**cfiles** ファイルにリストされる名前によって、コマンドは以下のようになります。

```
lint -a main.c readit.c gettoken.c . . .
lint -a getisx.c getprp.c getpid.c . . .
lint -a fltadd.c fltmult.c fltdiv.c . . .
```

このコマンド・シーケンスは、一度にすべてのファイル名を使用して **lint** コマンドを実行する場合とは少し異なります。**lint** コマンドはファイル間の相互参照を検査します。しかし、この例では、**main.c** と **fltadd.c** の間の、または別個のコマンド・ラインにリストされた 2 つのファイルの間を検査できません。

このため、すべてのファイル名が 1 行に入りきる場合にのみ、コマンドを実行したい場合があります。そのような場合は、以下のように **-x** フラグを使用して、これを **xargs** コマンドに対して指定します。

```
xargs -x lint -a <cfiles
```

cfiles ファイル内のすべてのファイル名が 1 つのコマンド・ラインに入りきらない場合には、**xargs** コマンドはエラー・メッセージを表示します。

2. ある数個のファイル名が入っているコマンドを作成するには、以下のように入力します。

```
xargs -t -n 2 diff <<EOF
starting chap1 concepts chap2 writing
chap3
EOF
```

このコマンド・シーケンスは、それぞれ 2 つのファイル名を含む (**-n 2**) **diff** コマンドを作成して、実行します。

```
diff starting chap1
diff concepts chap2
diff writing chap3
```

-t フラグを使用すると、**xargs** コマンドは実行する前に各コマンドを表示するので、現在何をしているか見ることができます。<<EOF と EOF パターン・マッチング文字は、最終行の前に入力されたテキストを **xargs** コマンドの標準入力として使用する *here* ドキュメントを定義します。

3. コマンド・ラインの中間にファイル名を挿入するには、以下のように入力します。

```
ls | xargs -t -I {} mv {} {}.old
```

このコマンド・シーケンスは、各名前最後に **.old** を追加することで、現行ディレクトリー内のすべてのファイルを名前変更します。**-I** フラグは、**xargs** コマンドに対して、{} (中括弧) が表示される所に **ls** ディレクトリー・リストの各行を挿入するように指示します。現行ディレクトリーにファイル **chap1**、**chap2**、**chap3** が入っている場合、このコマンドは以下のコマンドを作成します。

```
mv chap1 chap1.old
mv chap2 chap2.old
mv chap3 chap3.old
```

4. 個別に選択したファイル上でコマンドを実行するには、以下のように入力します。

```
ls | xargs -p -n 1 ar r lib.a
```

このコマンド・シーケンスによって、ライブラリー **lib.a** に追加するファイルを選択できます。**-p** フラグは、**xargs** コマンドに対して、作成する各 **ar** コマンドを表示し、実行したいかどうかを尋ねるプロンプトを表示するように指示します。コマンドを実行するには、**y** を入力します。コマンドを実行しない場合は、それ以外の任意のキーを押します。

以下のような画面が表示されます。

```
ar r lib.a chap1 ?...
ar r lib.a chap2 ?...
ar r lib.a chap3 ?...
```

5. 特定の数の引数が入ったコマンドを作成したり、それらの引数をコマンド・ラインの中に挿入するには、以下のように入力します。

```
ls | xargs -n6 | xargs -I{} echo {} - some files in the directory
```

現行ディレクトリーにファイル chap1 から chap10 が入っている場合は、以下のように出力されます。

```
chap1 chap2 chap3 chap4 chap5 chap6 - some files in the directory
chap7 chap8 chap9 chap10 - some files in the directory
```

ファイル

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

関連情報:

ar コマンド

diff コマンド

lint コマンド

xauth コマンド

目的

X サーバーへの接続で使用される権限情報を編集および表示します。

構文

```
xauth [ -f AuthFile ] [ -v | -q ] [ -i ] [ -b ] [ CommandArgument ... ]
```

説明

xauth コマンドは通常、X サーバーへの接続に使用される権限情報を編集および表示する際に使用します。このプログラムは、1 つのコンピューターから許可レコードを抽出し、他のコンピューターのレコードと組み合わせます (例えば、リモート・ログインを使用するときや、他のユーザーへのアクセスを認可するとき)。

下記のコマンドを、**xauth** コマンド・ライン上、またはスクリプト中に、対話形式で入力することができます。このプログラムは、X サーバーには接続しないことに注意してください。

項目	説明
add <i>DisplayName ProtocolName Hexkey</i>	所定のプロトコルおよびキー・データを使用する指定されたディスプレイのための許可エントリーが、許可ファイルに追加されます。データは、16 進数字の等しい長さの文字列として指定され、おのおのの組は、1 つの 8 進数を表します。おのおのの組の第 1 桁は、8 進数の上位 4 ビットを与え、第 2 桁は、下位 4 ビットを与えます。例えば、32 文字の 16 進キーは、128 ビットの値を表します。単一のピリオドのみで構成されているプロトコル名は、 MIT-MAGIC-COOKIE-1 の省略語として扱われます。

項目

extract *FileName DisplayName...*

generate *DisplayName ProtocolName [trusted | untrusted]*
[timeout seconds] [group group-id] [data hexdata]

list [*DisplayName...*]

merge [*FileName...*]

説明

指定された各ディスプレイの許可エントリーは、指定のファイルに書き込まれます。抽出されたエントリーは、**merge** および **nmerge** コマンドを使用して読み返すことができます。ファイル名に単一のハイフンのみを指定した場合、エントリーはバイナリ出力に書き込まれます。

このコマンドは **add** と類似しています。主な相違点は、ユーザーにキー・データの提供を要求するのではなく、キー・データを入手して権限ファイルに保管するために、*displayname* で指定されたサーバーに接続して SECURITY 拡張機能を使用することです。サーバーに接続できないか、または SECURITY 拡張機能がサポートされない場合、コマンドは失敗します。そうでない場合は、所定のプロトコルを使用して指定されたディスプレイのための権限エントリーが、権限ファイルに追加されます。単一のピリオドのみで構成されているプロトコル名は、MIT-MAGIC-COOKIE-1 の省略語として扱われます。

トラステッド・オプションを使用すると、この権限を使用して接続するクライアントは、通常どおりすべて表示されます。非トラステッドを使用する場合は、この権限を使用して接続するクライアントは、非トラステッドと見なされ、トラステッド・クライアントに属するデータの盗用や悪用が防止されます。非トラステッド・クライアントに課される制約事項の詳細については、SECURITY 拡張機能の仕様を参照してください。デフォルトは非トラステッドです。

タイムアウト・オプションは、この権限が有効となる期間を秒単位で指定します。権限がこの期間よりも長く使用されないままになっている (クライアントが接続されない) 場合、サーバーがこの権限を除去するので、このあとこれを使用して接続しようとしても失敗します。サーバーが行う除去では、権限エントリーは権限ファイルから削除されないことに注意してください。デフォルトのタイムアウトは 60 秒です。

グループ・オプションは、この権限に接続中のクライアントが属すべきアプリケーション・グループを指定します。詳細については、アプリケーション・グループ拡張機能の仕様を参照してください。デフォルトは「アプリケーション・グループに属していない」です。

データ・オプションは、権限を生成するためにサーバーが使用するデータを指定します。これは、権限ファイルに書き込まれたデータと同じデータではないことに注意してください。このデータの変換処理は、権限プロトコルによって決まります。*hexdata* の形式は、**add** コマンドに記述される *hexkey* と同じです。デフォルトは「データを送信しない」です。

指定された各ディスプレイ (ディスプレイが指定されていない場合には、すべてのディスプレイ) の許可エントリーを、標準出力上にテキスト・フォーマットで出力します。キー・データは常に、**add** コマンドの記述の中で与えられた 16 進数フォーマットで表示されます。

指定されたファイルから許可エントリーが読み取られ、許可データベースの中に組み込まれます。このとき、既存のエントリーと一致するものをすべて置き換えます。ファイル名に単一のハイフンだけを指定した場合、まだ読み取られていなければ、バイナリ入力が読み取られます。

項目	説明
[n]extract <i>FileName DisplayName...</i>	指定された各ディスプレイの許可エントリーは、指定のファイルに書き込まれます。エントリーは、(セキュア電子メールなどの)非バイナリー転送に適した数値フォーマットで書き込まれます。抽出されたエントリーは、 merge および nmerge コマンドを使用して読み返すことができます。ファイル名に単一のハイフンのみを指定した場合、エントリーは標準出力に書き出されます。指定された各ディスプレイ (ディスプレイが指定されていない場合には、すべてのディスプレイ) の許可エントリーを nextract コマンドで使用される数値フォーマットで標準出力上に出力します。キー・データは常に、 add コマンドの記述の中で与えられた 16 進数フォーマットで表示されます。
[n]list [<i>DisplayName...</i>]	指定された各ディスプレイ (ディスプレイが指定されていない場合には、すべてのディスプレイ) の許可エントリーを nextract コマンドで使用される数値フォーマットで標準出力上に出力します。キー・データは常に、 add コマンドの記述の中で与えられた 16 進数フォーマットで表示されます。
[n]merge [<i>FileName...</i>]	指定されたファイルから許可エントリーが読み取られ、許可データベースの中に組み込まれます。このとき、既存のエントリーと一致するものをすべて置き換えます。 extract コマンドの記述の中で与えられた数値フォーマットが使用されます。ファイル名に単一ハイフンのみを指定した場合、まだ読み取られていなければ、標準入力を読み取られます。
remove <i>DisplayName...</i>	指定されたディスプレイと一致する許可エントリーは、許可ファイルから除去されます。
source <i>FileName</i>	指定されたファイルは、実行する xauth コマンドを含むスクリプトとして取り扱われます。ブランク行および # 文字 (ポンド記号) で始まる行は無視されます。単一のハイフンは、まだ読み取られていなければ、標準入力を指定するのに使用できます。
info	許可ファイル、変更の有無、および xauth コマンドが読み取られている場所についての記述情報を、標準出力上に出力します。
exit	変更が起きた場合、(許可されれば) 許可ファイルに書き込まれ、プログラムは終了します。ファイルの終わりは、暗黙的に終了コマンドとして処理されます。
quit	どのような変更も無視して、プログラムを終了します。割り込み文字を押すことによっても、プログラムを終了することができます。
help [<i>String</i>]	指定された文字列で始まるコマンドすべての (文字列が指定されていない場合にはすべてのコマンドの) 説明を、標準出力上に出力します。
?	有効コマンドの簡潔なリストを標準出力上に出力します。

add、**[n]extract**、**[n]list**、**[n]merge**、**remove** コマンドのディスプレイ名は、**DISPLAY** 環境変数および共通ディスプレイ・コマンド・ライン引数と同じフォーマットを使用します。(画面番号などの) ディスプレイに固有な情報は必要とされず、無視されます。同一機接続 (例えば、ローカル・ホスト・ソケット、共用メモリー、およびインターネット・プロトコル *HostName*) は、*HostName/unix:DisplayNumber* として参照され、異なるコンピューターのローカル・エントリーを 1 つの許可ファイルに保管できるようになっています。

注: 安全とはいえないネットワークを持っているユーザーは、コンピューター間の許可エントリーをコピーするために暗号化されたファイル転送機構を使用するときは、注意が必要です。同様に、MIT-MAGIC-COOKIE-1 プロトコルは、セキュリティーのない環境では、あまり役立つものではありません。よりセキュリティーを重視するなら、そのようなサイトでは、Kerberos などの暗号化された許可機構を使用する必要があります。現在のところ、スペースをプロトコル名に入れることはできません。引用符を加えることはできます。

フラグ

下記のようなオプションは、**xauth** コマンドとともに使用します。これらは、個別に (例えば **-q -i** のように) 指定することも、または結合して (例えば **-qi**) のように) 指定することもできます。

項目	説明
-f <i>AuthFile</i>	使用する許可ファイルの名前を指定します。デフォルトでは、 xauth は XAUTHORITY 環境変数、あるいはユーザーのホーム・ディレクトリーにある .xauthority が指定するファイルを使用します。
-v	xauth が詳細モードで動作し、さまざまな操作の結果 (例えば、読み取りまたは書き込みされたレコードの数) を示す状況メッセージを出力することを示します。 xauth コマンドが標準入力からコマンドを読み取り、標準出力を端末に送信する場合には、これがデフォルトとなります。
-q	xauth が抑制モードで動作し、不必要な状況メッセージを出力しないことを示します。コマンド・ラインに xauth コマンドが指定された場合、または標準出力を端末に送信しない場合は、これがデフォルトとなります。
-i	xauth がどのような許可ファイル・ロックも無視しなければならないことを示します。通常、 xauth は、他のプログラム (一般には xdm または他の xauth) によってロックされた許可ファイルの読み取り、または編集は、どのようなものでも拒否します。
-b	xauth がどのような許可ファイル・ロックにも、その処理を行う前に中断を試行しなければならないことを示します。古くなったロックをクリーンアップする場合にのみ使用します。

例

下記のように、現行ディスプレイ用のエントリーを抽出して別のコンピューターへコピーし、リモート・マシン上でユーザーの許可ファイルと組み合わせるのが、**xauth** コマンドの最も一般的な使用方法です。

```
% xauth extract ¥- $DISPLAY | rsh otherhost xauth merge ¥-
```

ファイル

項目	説明
\$HOME/.Xauthority	XAUTHORITY 環境変数が定義されていない場合のデフォルトの許可ファイルが入っています。

xclock コマンド

目的

現在の時刻を継続的に表示します。

構文

```
xclock [ -Xtoolkitoption ... ] [ -analog | -digital ] [ -chime ] [ -hd Color ] [ -help ] [ -hl Color ] [ -padding Number ] [ -update Seconds ]
```

説明

xclock コマンドは、システム・クロックから時刻を取り出して、この時刻をデジタル時計またはアナログ時計のフォーマットで表示し更新します。クロックをアナログ・フォーマットまたはデジタル・フォーマットで表示するには、**-analog** または **-digital** フラグを選択します。また、フラグを選択して、チャイムと更新頻度、カラー、およびボーダー幅など、クロックの表示状態を指定することもできます。

このコマンドは、Athena クロック・ウィジェットを使用します。Athena では、コアとなるリソース名とクラスが認識されます。これらのリソースを指定するには、**xclock** コマンドを構成するウィジェットの階層を認識しておく必要があります。下記の例で、字下げされた項目は階層構造を示します。ウィジェットのクラス名が最初に表示され、その後ウィジェット・インスタンス名が表示されています。

```
XClock xclock
Clock clock
```

下記の例は、このクライアントに対するリソースを指定するための、可能な方法を示しています。

xclock.clock.background
XClock*background
xclock*background

注: xclock.background としてリソースを指定することは、以前のバージョンでは機能していましたが、このバージョンでは機能しません。

フラグ

項目	説明
-Xtoolkitoption	xclock コマンドには、リストされている特定のフラグの他に、標準の X Toolkit コマンド・ライン・オプション・フラグをすべて使用できます。 「 custom コマンド」で、標準の X Toolkit コマンド・ライン・オプション・フラグを表示できません。
-analog	デフォルト・モードであるアナログ表示モードを設定します。各分ごとの目盛りと時間ごとのストローク・マークが付いている従来の型の 12 時間クロックの文字盤を表示します。
-chime	30 分に一度、つまり 1 時間に 2 度、チャイムを鳴らすように設定します。
-digital	24 時間のデジタル表示モードを設定します。デジタル・フォーマットで日付と時刻を表示します。
-hd Color	カラー・ディスプレイ上でアナログ・モードによる針の色を指定します。デフォルトは黒です。
-help	使用可能なオプションの簡単な要約を出力します。
-hl Color	(HL の小文字) アナログ時計の針の先端の強調表示の色を指定します。デフォルトは黒です。
-padding Number	ウィンドウのボーダーとクロック・テキスト、あるいはピクチャーの間の埋め込みの幅をピクセル単位で指定します。デフォルトは 8 です。
-update Seconds	xclock コマンドが表示を更新する頻度を秒単位で指定します。 xclock ウィンドウが覆われた後で露出すると、 xclock コマンドは即座に再表示を行います。30 秒未満の更新頻度を指定すれば、アナログ・モードで秒針が使用可能になります。デフォルトの更新頻度は 60 秒です。

.Xdefaults キーワード

xclock コマンドのデフォルトを設定するには、下記のキーワードを使用します。

項目	説明
analog (Boolean クラス)	デジタル時計の代わりにアナログ時計を指定します。デフォルトは True です。
chime (Boolean クラス)	ベルを正時と 30 分に鳴らすかどうかを指定します。
fontSet (FontSet クラス)	デジタル時計のフォント・セットを指定します。可変幅のフォントは、常に正しく表示されるとは限りません。
foreground (Foreground クラス)	カラー・ディスプレイ上の目盛りマークの色を指定します。 reverseVideo を指定するとデフォルトは白になり、それ以外の場合のデフォルトは黒になります。
hands (Foreground クラス)	カラー・ディスプレイ上のアナログ時計の針の内側の色を指定します。 reverseVideo を指定するとデフォルトは白になり、それ以外の場合のデフォルトは黒になります。
highlight (Foreground クラス)	時計の針の強調表示に使用する色を指定します。 reverseVideo を指定するとデフォルトは白になり、それ以外の場合のデフォルトは黒になります。
height (Height クラス)	時計の高さを指定します。アナログ時計のデフォルトは 164 ピクセルです。デジタル時計のデフォルトは、選択したフォントで時計を表示するときに必要なピクセル数です。
padding (Margin クラス)	使用される内部埋め込みの数量をピクセル単位で指定します。デフォルトは 8 です。
update (Interval クラス)	xclock コマンドの表示の更新頻度を秒単位で指定します。
width (Width クラス)	時計の幅を指定します。アナログ時計のデフォルトは 164 ピクセルです。デジタル時計のデフォルトは、選択したフォントで時計を表示するときに必要なピクセル数です。

環境変数

項目	説明
DISPLAY	デフォルトのホスト番号およびディスプレイ番号を取り出します。
XENVIRONMENT	RESOURCE_MANAGER 属性に保管されているグローバル・リソースを上書きするリソース・ファイルの名前を取り出します。

例

1. デジタル時計表示を指定するには、下記のように入力します。

```
xclock -digital
```

2. アナログ時計に赤い針を指定するには、下記のように入力します。

```
xclock -hd red
```

ファイル

項目	説明
<code>/usr/lib/X11/app-defaults/XClock</code>	必要なリソースを指定します。

関連情報:

custom コマンド

xcmsdb コマンド

目的

画面のルート・ウィンドウの属性に保管されたスクリーン・カラー特性データのロード、照会、および除去を行います。

注: **xcmsdb** コマンドは、X11R5 (AIXwindows Version 1.2.3) でのみサポートされます。

構文

```
xcmsdb [ -display Display ] [ [ -query ] [ -remove ] [ -color ] ] | [ -format 32 | 16 | 8 ] [ FileName ]
```

説明

xcmsdb コマンドは画面のルート・ウィンドウの属性に保管されたスクリーン・カラー特性データのロード、照会、および除去に使用します。スクリーン・カラー特性データは、デバイス独立およびデバイス依存のカラー指定を相互に適切に変換するために必要な **Xlib** の統合部です。 **Xlib** は **XDCCC_LINEAR_RGB_MATRICES** と **XDCCC_LINEAR_RGB_CORRECTION** の各属性を使って、カラー・モニター用のカラー特性データを保管します。また、グレースケール・モニター用には **XDCCC_GRAY_SCREENWHITEPOINT** と **XDCCC_GRAY_CORRECTION** の各属性を使用します。 **Xlib** はスクリーン・カラー特性機能セットの追加が可能であるため、追加された機能セットが他の属性上にスクリーン・カラー特性データを配置することがあります。このユーティリティーは、他の属性かどうかを認識しないため、機能セットのある適切なユーティリティーを使用するか、サンプルの **xprop** ユーティリティーを使用する必要があります。

-query または **-remove** フラグ・オプションが指定されていない場合、*FileName* パラメーター (入力ファイルが指定されていない場合は標準入力) の ASCII 形式で読み取り可能な内容は、属性に保管できるように適切に変換されます。

注: **libX11.a** 内の Xcms API はサポートされますが、サポートされないサンプルとして、クライアント側のカラー名データベース */usr/lib/X11/Xcms.txt*、およびデバイス・カラー特性ファイル */usr/lib/X11/XcmsIBM5081.dcc* が提供されています。

フラグ

項目	説明
-display <i>Display</i>	変換先のサーバーを指定します。
-query	画面のルート・ウィンドウの XDCCC 属性を読み取るか、または読み取ろうとします。読み取りが成功した場合、データをより読み取り可能なフォーマットに変換し、標準出力に送信します。
-remove	画面のルート・ウィンドウの XDCCC 属性を除去するか、または除去しようとしています。
-color	XDCCC_LINEAR_RGB_MATRICES および XDCCC_LINEAR_RGB_CORRECTION の各属性を検査するだけの目的で、 -query および -remove オプションを設定します。 -color オプションが設定されていない場合、 -query および -remove オプションはすべての属性を検査します。
-format 32 16 8	XDCCC_LINEAR_RGB_CORRECTION 属性に対して、属性フォーマット (1 エントリーにつき 32、16、8 ビットのいずれか) を指定します。1 エントリーに対するビットの値が増えると、エンコードされた浮動小数点値の精度が上がります。デフォルトは、1 エントリーにつき 32 ビットです。

パラメーター

項目	説明
<i>FileName</i>	スクリーン・カラー特性データ・ファイルの ASCII 形式で読み取り可能な内容を指定します。

例

- 下記の例は、ファイルからスクリーン・カラー特性データを読むように **xcmsdb** コマンドに指示して、これをルート・ウィンドウ上に置く方法です。

```
xcmsdb /usr/lib/X11/XcmsIBM5081.dcc
```

- 下記の例は、スクリーン・カラー特性データをルート・ウィンドウ上に置いた後で、そのデータが存在すれば読み返すよう、**xcmsdb** コマンドに指示しています。

```
xcmsdb -query
```

xdm コマンド

目的

XDMCP をサポートする X Display の集合を管理します。

構文

```
xdm [ -config ConfigurationFile ] [ -debug DebugLevel ] [ -nodaemon ] [ -error ErrorLogFile ] [ -resources ResourceFile ] [ -server ServerEntry ] [ -udpPort PortNumber ] [ -session SessionProgram ] [ -xrm ResourceSpecification ]
```

説明

xdm (X Display Manager) コマンドは、ローカル・ホストまたはリモート・サーバー上にある X Display の集合を管理します。**xdm** コマンドの設計は、X 端末の必要性および X コンソーシアム規格 XDMCP (X Display Manager 制御プロトコル) に準拠しています。**xdm** コマンドが提供するサービスは、文字端末における **init**、**getty**、および **login** コマンドが提供するサービスと似ています。つまり、**xdm** コマンドは、ログイン名とパスワードの入力を求め、ユーザーを認証し、セッションを実行するというサービスを提供します。

セッション は、特定のプロセスの存続期間によって定義されます。つまり、従来の文字ベースの端末の世界では、ユーザーのログイン・シェルのことです。しかし、**xdm** の世界では、任意のセッション・マネージャーを指します。なぜなら、ウィンドウ操作環境では、ユーザーのログイン・シェル・プロセスは、接続に端末のようなインターフェースを持つ必要がないからです。実際のセッション・マネージャーが使用できない場合は、通常、ウィンドウ・マネージャーや端末エミュレーターがセッション・マネージャーとして使用されます。つまり、このプロセスの終了がユーザーのセッションの終了を意味します。

セッションが終了すると、**xdm** は X サーバーをリセットし、(必要であれば) プロセス全体を再始動します。

xdm コマンドは、XDMCP 経由で **Indirect** 照会を受信すると、**chooser** プロセスを実行して、ディスプレイの代わりに XDMCP の **BroadcastQuery** を (指定されたホストには XDMCP Query) を実行し、XDMCP ディスプレイ管理機能を提供できるホストのメニューを表示します。この機能は、ホスト・メニューを提供しない X 端末に役立ちます。

xdm コマンドはユーザーが最初に目にするインターフェースを提供するため、使いやすく、特定のサイトの必要に応じて簡単にカスタマイズできるように設計されています。

一般的な使用方法

xdm コマンドは、さまざまな環境で操作できるように設計されています。

最初に、**xdm** 構成ファイルをセットアップします。次に、関連するすべてのファイルを入れるディレクトリー (通常は、**/usr/lib/X11/xdm**) を作成します。下記は、構成ファイルの例です。**xdm-config** という名前を付けることにします。

```
DisplayManager.servers:      /usr/lib/X11/xdm/Xservers
DisplayManager.errorLogFile: /usr/lib/X11/xdm/xdm-errors
DisplayManager*resources:    /usr/lib/X11/xdm/Xresources
DisplayManager*startup:      /usr/lib/X11/xdm/Xstartup
DisplayManager*session:      /usr/lib/X11/xdm/Xsession
DisplayManager.pidFile:      /usr/lib/X11/xdm/xdm-pid
DisplayManager._0.authorize: true
DisplayManager*authorize:    false
```

このファイルには、他のファイルへの参照が含まれます。いくつかのリソースは、コンポーネントを区切る * (アスタリスク) で指定されています。これらのリソースは、各ディスプレイに対して固有に作成することができ、* (アスタリスク) の部分にディスプレイの名前が入ります。しかし通常はあまり役には立ちません。詳細は、次のページのリソースのセクションを参照してください。

最初のファイル **/usr/lib/X11/xdm/Xservers** には、XDMCP を使用しないで管理するディスプレイのリストが入っています。ほとんどのワークステーションは 1 つのディスプレイ (番号 0 (ゼロ) が割り当てられている) しか持っていないため、このファイルは下記のような内容になります。

```
:0 Local local /usr/bin/X11/X -force
```

これにより、`/usr/bin/X11/X` が継続して当該ディスプレイ上で動作し、連続したセッションのサイクルを管理します。

`/usr/lib/X11/xdm/xdm-errors` ファイルには、`xdm` のエラー・メッセージ、および `Xsetup`、`Xstartup`、`Xsession`、`Xreset` のいずれかのスクリプトが標準エラーに書き込んだ出力が入っています。`xdm` コマンドの始動に何らかの問題が発生した場合は、`/usr/lib/X11/xdm/xdm-errors` ファイルを検査して、`xdm` コマンドに原因があるかどうか確認します。

構成ファイルの次の項目 `/usr/lib/X11/xdm/Xresources` は、`xrdb` コマンドによって、リソース・データベースとしてディスプレイにロードされます。認証ウィジェットは始動する前にこのデータベースを読み取るため、構成ファイルには通常、認証ウィジェット用のパラメーターが入っています。

フラグ

オプションで指定する値はすべて (`-config` で指定するものは除く)、構成ファイルの中でリソースとしても指定できます。

項目	説明
<code>-config</code> <i>ConfigurationFile</i>	<code>xdm</code> コマンドの動作を制御するリソースを指定する構成ファイルに名前を付けます。デフォルトは、 <code>/usr/lib/X11/xdm/xdm-config</code> ファイルです。
<code>-debug</code> <i>DebugLevel</i>	<code>DisplayManager.debugLevel</code> リソースに対する数値を指定します。ゼロ以外の値を指定すると、 <code>xdm</code> はデバッグ・ステートメントを端末に出力し、 <code>DisplayManager.daemonMode</code> リソースを使用不可にし、強制的に <code>xdm</code> を同期実行するようにします。これらのエラー・メッセージはわかりにくいこともあります。このような場合、 <code>xdm</code> コマンドの <code>X11R4</code> ソース・コードを検査して、エラーの意味を調べてください。
<code>-nodaemon</code>	<code>DisplayManager.daemonMode</code> リソースに対する値として <code>False</code> (偽) を指定します。これによって、デーモンの通常の動作を抑制します。この動作により <code>xdm</code> コマンドは最初の始動時にすべてのファイル・ディスクリプターをクローズし、制御端末との関連付けを無効にし、自分の実行をバックグラウンドに移します。
<code>-error</code> <i>ErrorLogFile</i>	<code>DisplayManager.errorLogFile</code> リソースに対する値を指定します。このファイルには、 <code>xdm</code> のエラーと、セッションの進行中に実行されたさまざまなスクリプトとプログラムが標準エラーに書き込んだ出力が入っています。
<code>-resources</code> <i>ResourceFile</i>	<code>DisplayManager*resources</code> リソースに対する値を指定します。このファイルは、 <code>xrdb</code> コマンドによってロードされ、認証ウィジェットの構成パラメーターを指定します。
<code>-server</code> <i>ServerEntry</i>	<code>DisplayManager.servers</code> リソースに対する値を指定します。サーバー仕様のセクションには、このリソースについての説明があります。
<code>-udpPort</code> <i>PortNumber</i>	<code>DisplayManager.requestPort</code> リソースに対する値を指定します。これにより、 <code>xdm</code> コマンドが <code>XDMCP</code> 要求をモニターするポート番号が設定されます。 <code>XDMCP</code> は、登録されている定義済 UDP ポート 177 を使用します。デバッグするとき以外、このリソースは変更しないでください。
<code>-session</code> <i>SessionProgram</i>	<code>DisplayManager*session</code> リソースに対する値を指定します。このリソースによって指定されるプログラムは、ユーザーのログイン後にセッションとして実行されます。
<code>-xrm</code> <i>ResourceSpecification</i>	ほとんどの X ツールキット・アプリケーションと同様、任意のリソースを指定できます。

リソース

X リソース・フォーマットの構成ファイルを使って、`xdm` のアクションをさまざまな段階において制御することができます。`xdm` のアクションをすべてのディスプレイで変更するリソースもあれば、単一のディスプレイで変更するリソースもあります。特定のディスプレイにアクションが関連する場合、リソース名の「`DisplayManager`」と最後のリソース名セグメントの間にディスプレイ名が挿入されます。例えば、`DisplayManager.expo_0.startup` は、「`expo:0`」ディスプレイ上の始動シェル・ファイルを定義するリソース名です。リソース・マネージャーは、リソース名とその値をコロンで区切り、リソース名の部分をドット

で区切っています。しかし、**xdm** がリソース名を生成するときには、ドットとコロンの両方を下線に置き換えます。

項目	説明
DisplayManager.servers	複数のサーバー・エントリーから構成されるファイル名 (値がスラッシュで始まる場合は 1 行に 1 エントリー) を指定します。または、単一のサーバー・エントリーのファイル名を指定します。詳細については、サーバー仕様のセクションを参照してください。
DisplayManager.requestPort	xdm コマンドが XDMCP の着信要求を listen するために使用する UDP ポート番号を示します。システムをデバッグする必要がない限り、この値はデフォルトの 177 のままにしておいてください。
DisplayManager.errorLogFile	コンソールではなく、指定したファイルにエラー・メッセージをリダイレクトします。このファイルには、 Xsetup 、 Xstartup 、 Xsession 、および Xreset ファイルが標準エラーに送信した出力も入っています。したがって、上記スクリプトの例で発生した問題の解説もこのファイルに入ります。
DisplayManager.debugLevel	このリソースの整数値が 0 (ゼロ) より大きい場合、 xdm コマンドは大量のデバッグ情報を出力します。また、 xdm コマンドはデーモン・モードをオフにします (デーモン・モードはデバッグ情報を廃棄し、root 以外のユーザーも xdm コマンドを実行できるモードで、概して役に立ちません)。
DisplayManager.daemonMode	xdm コマンドは、どの端末にも関係のないデーモン・プロセスになろうとします。これは、親プロセスを fork および離脱することによって終了し、次にファイル・ディスクリプターをクローズし、最後に制御端末を解放することによって行われます。環境によっては (特に、デバッグ中は)、このモードが望ましくないことがあります。このような場合、このリソースに False (偽) を設定し、この機能を使用不可にしてください。
DisplayManager.pidFile	指定したファイルが作成され、主 xdm プロセスのプロセス ID が ASCII 表記で入ります。また、 xdm コマンドはファイル・ロック機能はこのファイルに使用し、同一コンピュータで複数のデーモンが実行されることを防ぎます (予期せぬ結果を引き起こすからです)。
DisplayManager.lockPidFile	複数 Display Manager の同時実行を防ぐファイル・ロック機能を xdm コマンドが使用するかどうかを制御します。
DisplayManager.authDir	セッションの初期化中に xdm コマンドが許可ファイルを保管するディレクトリーを指定します。デフォルトは、 /usr/lib/X11/xdm です。
DisplayManager.autoRescan	セッションが終了し、ファイルが変更された後で、 xdm コマンドが構成、サーバー、アクセス制御、および認証キーの各ファイルを再スキャンするかどうかをブール値で決定します。デフォルトは True (真) です。主プロセスに SIGHUP シグナルを送信することによって、 xdm デーモンに再度上記ファイルを読み取らせることができます。
DisplayManager.removeDomainname	XDMCP クライアント用のディスプレイ名を計算するとき、通常、ネーム・リゾルバーは端末に対して完全に修飾された端末のホスト名を作成します。これは紛らわしいこともあります。このような場合、この変数を設定しておけば、ホスト名のドメイン・ネームがローカル・ホストのドメイン・ネームと同じ場合、 xdm コマンドはホスト名のドメイン・ネームの部分を除去します。デフォルト値は True です。
DisplayManager.keyFile	XDM-AUTHENTICATION-1 スタイルの XDMCP 認証は、秘密鍵を xdm デーモンと端末の間で共用することを要求します。このリソースには、このような値が入ったファイルを指定します。このファイルの各エントリーは、ディスプレイ名と共用キーから構成されます。デフォルトでは、 xdm コマンドは XDM-AUTHENTICATION-1 をサポートしません。なぜなら、 XDM-AUTHENTICATION-1 に必要なデータ暗号化規格 (DES) は、アメリカ合衆国の輸出制限のため、通常は配布できないからです。

項目	説明
DisplayManager.accessFile	許可されていない XDMCP サービスを防ぎ、XDMCP の IndirectQuery 要求の転送を許可するために、このファイルには、当該コンピューターへの直接アクセスが認められている、照会の転送先のホストのリストを持つホスト名のデータベースが入っています。このファイルのフォーマットについては、XDMCP アクセス制御のセクションを参照してください。
DisplayManager.exportList	Xsetup 、 Xstartup 、 Xsession 、および Xreset プログラムに渡される追加の環境変数のリストで、このリストは、ホホワイト・スペースで区切られています。
DisplayManager.randomFile	許可キーのシード (seed) を生成するためにチェックサムを行うためのファイルです。これは、頻繁に変更されるファイルである必要があります。デフォルトは、 /dev/mem です。
DisplayManager.choiceTimeout	ユーザーが chooser からホストを選択した後、応答が表示されるまでに待機する時間を秒数で指定します。ディスプレイがこの間に XDMCP IndirectQuery を送信する場合、要求は、選択されたホストへ転送されます。そうでない場合には、新規のセッションからの要求と見なされ、 chooser がもう一度提供されます。デフォルトは 15 です。
DisplayManager.DISPLAY.resources	xrdb コマンドによってリソース・データベースとしてディスプレイの画面 0 のルート・ウィンドウにロードされるファイルの名前を指定します。ログイン・ウィジェット、 Xsetup 、および chooser プログラムが、このファイルのリソースを使用します。このリソース・データベースは、認証プロシージャが開始される直前にロードされます。したがって、このリソース・データベースは、ログイン・ウィンドウの外観を制御することができます。認証クライアントのセクションには、このファイルに入れるべきさまざまなリソースについての説明があります。このリソースにはデフォルト値はありませんが、 /usr/lib/X11/xdm/Xresources が標準的に使用されます。
DisplayManager.DISPLAY.chooser	特殊なホスト名 CHOOSER にリダイレクトされた間接照会用のホスト・メニューを提供するために実行するプログラムを指定します。デフォルトは、 /usr/lib/X11/xdm/chooser です。XDMCP アクセス制御のセクションと Chooser のセクションを参照してください。
DisplayManager.DISPLAY.xrdb	リソースを読み込むために使用するプログラムを指定します。デフォルトでは、 xdm コマンドは /usr/bin/X11/xrdb を使用します。
DisplayManager.DISPLAY.cpp	xrdb コマンドが使用する C プリプロセッサの名前を指定します。ログイン・ウィンドウを提供する前に (root として) 実行するプログラムを指定します。このリソースは、ログイン・ウィンドウに関連するスクリーンの外観を変更するときや他のウィンドウを立ち上げるときに使用します (例えば、ここで xconsole を実行するなどです)。デフォルトで実行するプログラムはありません。通常ここで使用されるファイルは Xsetup です。セットアップ・プログラムのセクションを参照してください。
DisplayManager.DISPLAY.startup	認証プロセスの成功後に (root として) 実行するプログラムを指定します。デフォルトで実行するプログラムはありません。通常ここで使用されるファイルは Xstartup です。始動プログラムのセクションを参照してください。
DisplayManager.DISPLAY.session	実行するセッションを指定します (root としては実行しない場合)。デフォルトでは、 /usr/bin/X11/xterm が実行されます。通常ここで使用される名前は Xsession スクリプトです。セッション・プログラムのセクションを参照してください。
DisplayManager.DISPLAY.reset	セッションの終了後に (root として) 実行するプログラムを指定します。デフォルトで実行するプログラムはありません。通常ここで使用される名前は Xreset スクリプトです。リセット・プログラムのセクションを参照してください。
DisplayManager.DISPLAY.openDelay	非一過性のサーバーをオープンしようとする際、連続するオープンの間に一時停止する時間の長さ (秒数) を指定することによって、 xdm コマンドの動作を制御します。

項目 DisplayManager.DISPLAY.openRepeat	説明 非一過性のサーバーをオープンしようとする際、オープンを試みる回数を指定することによって、 xdm コマンドの動作を制御します。
DisplayManager.DISPLAY.openTimeout	非一過性のサーバーをオープンしようとする際、実際にオープンを試みる間待つ秒数 (つまり、 connect(2) システム・コールで費やされる最大時間) を指定することによって、 xdm コマンドの動作を制御します。
項目 DisplayManager.DISPLAY.startAttempts	説明 非一過性のサーバーをオープンしようとする際、そのサーバーへの接続をあきらめるまでに繰り返されるプロセス全体の回数を指定し、 xdm コマンドの動作を制御します。Display Manager の、 openRepeat リソースに指定された回数の試行が終わると、または、Display Manager の
DisplayManager.DISPLAY.pingInterval	openTimeout リソースに指定された時間が経過するまで試行を繰り返すと、 xdm コマンドは終了し、そのサーバーを再始動し、再度接続しようとします。このプロセスは、 <i>startAttempts</i> で指定された回数だけ繰り返されます。そして、このときディスプレイはアクティブでないと宣言され、使用不可になります。この動作は一見任意の動作に見えますが、十分な経験とともに開発されたもので、ほとんどのシステムで有効です。 <i>openDelay</i> のデフォルトは 5 、 <i>openRepeat</i> のデフォルトは 5 、 <i>openTimeout</i> のデフォルトは 30 、 <i>startAttempts</i> のデフォルトは 4 です。
DisplayManager.DISPLAY.pingTimeout	リモート・ディスプレイを見失った時間を特定するために、 xdm コマンドは X 接続と XSync 呼び出しを使って定期的リモート・ディスプレイとの接続を確認 (ping) します。このリソースには、接続を確認する (ping) 間隔 (分) を指定します。デフォルトは 5 分です。管理ホストから独立した形態で X 端末を頻繁に使用する場合は、この値を増やしてください。 注: 端末が不意に使用不可になった後でも、AIXwindows セッションが存在し続けることがあります。 xdm コマンドは、ローカルのディスプレイの接続については確認 (ping) しません。NFS サービスに対してサーバーがハングし、接続の確認 (ping) に応答しなくなると、ワークステーション・セッションが終了することがあります。
DisplayManager.DISPLAY.pingTimeout	リモート・ディスプレイを見失った時間を特定するために、 xdm コマンドは X 接続と XSync 呼び出しを使って定期的リモート・ディスプレイとの接続を確認 (ping) します。このリソースには、端末が要求に応答するまでの最大待ち時間 (分) を指定します。端末が応答しなかった場合、そのセッションはアクティブでないと宣言され、終了します。デフォルトは 5 分です。管理ホストから独立した形態で X 端末を頻繁に使用する場合は、この値を増やしてください。 注: 端末が不意に使用不可になった後でも、AIXwindows セッションが存在し続けることがあります。 xdm コマンドは、ローカルのディスプレイの接続については確認 (ping) しません。NFS サービスに対してサーバーがハングし、接続の確認 (ping) に応答しなくなると、ワークステーション・セッションが終了することがあります。
DisplayManager.DISPLAY.terminateServer	セッション終了時、X サーバーをリセットせずにキャンセルするかどうかを指定します。このオプションは、サーバーが時間の制限を超えて実行し続ける傾向があるとき、サーバーの実行時間を制限するために使用できます。デフォルト値は False です。
DisplayManager.DISPLAY.userPath	xdm コマンドがセッション用の PATH 環境変数に設定する値を指定します。この値は、コロんで区切ったディレクトリーのリストである必要があります。詳細については、 sh コマンド コマンド・リファレンスのこのセクションを参照してください。通常の設定は、 :/bin:/usr/bin:/usr/bin/X11:/usr/ucb です。このリソースのデフォルトは、構成時に、AIXwindows システム構成ファイルの DefaultUserPath リソースで指定することができます。

項目	説明
DisplayManager.DISPLAY.systemPath	<p>xdm コマンドはこのリソースの値に、始動スクリプトおよびリセット・スクリプト用の PATH 環境変数を設定します。このリソースのデフォルトは、構成時に、システム構成ファイルの DefaultSystemPath リソースで指定することができます。通常の設定は、/etc/bin/usr/bin/usr/bin/X11/usr/ucb です。このエントリーには、. (ピリオド) (現行ディレクトリー) が存在していないことに注意してください。ルートもこれに従います。つまり、「トロイの木馬」的なシステム浸透スキーマを回避することができます。</p>
DisplayManager.DISPLAY.systemShell	<p>xdm コマンドはこのリソースの値に、始動スクリプトおよびリセット・スクリプト用の SHELL 環境変数を設定します。デフォルトは /bin/sh です。</p>
DisplayManager.DISPLAY.failSafeClient	<p>デフォルトのセッションの実行に失敗した場合に xdm コマンドが制御を戻す先のプログラムを指定します。このプログラムは、引数なしで、そのセッションが使用するはずだった環境変数を使って実行されます (セッション・プログラムのセクションを参照してください)。デフォルトは /usr/bin/X11/xterm です。</p>
DisplayManager.DISPLAY.grabServer	<p>セキュリティの向上のため、xdm コマンドは、ログイン名とパスワードの読み取り中にサーバーとキーボードをグラブ (占有) します。</p>
DisplayManager.DISPLAY.grabTimeout	<p>grabServer リソースは、ログイン名およびパスワード名の読み取り中にサーバーをグラブする必要がある場合に指定します。 False (偽) が設定されている場合、キーボードのグラブが成功した後、サーバーのグラブは解除されます。それ以外の場合はセッションが開始される直前までサーバーのグラブは解除されません。デフォルト値は False です。</p> <p>grabTimeout リソースには、グラブが成功するまでに xdm コマンドが待つ最大時間を指定します。他のクライアントがサーバーをグラブしている場合、またはネットワーク待ち時間が長い場合は、グラブが失敗することがあります。このリソースのデフォルト値は 3 秒です。この値を大きくする場合は注意が必要です。ユーザーがディスプレイ上のウィンドウに似ていると思いをすることがあるためです。グラブが失敗すると、xdm コマンドはアクティブでなくなり、サーバーを再始動し (可能であれば)、セッションを再始動します。</p>
DisplayManager.DISPLAY.authorize	<p>authorize は、xdm コマンドがローカル・サーバー接続の許可を作成および使用するかどうかをコントロールするブール・リソースです。許可を使用する場合、xdm コマンドは、authName リソースの値として、ホワイト・スペースで区切られたリストで指示された許可機構を使用します。 XDMCP 接続は、サポートされる許可機構を動的に指定します。したがって、この場合 authName リソースは無視されます。</p>
DisplayManager.DISPLAY.authName	<p>authorize リソースがディスプレイに設定され、許可が使用できない場合、ユーザーのログイン・ウィジェットに別のメッセージが表示されます。デフォルトでは、authorize リソースは True (真)、authName は MIT-MAGIC-COOKIE-1 です。</p>
DisplayManager.DISPLAY.authFile	<p>xdm コマンドとサーバーが -auth サーバー・コマンド・ライン・オプションを使って許可データの通信に使用するファイルを指定します。このファイルは簡単に除去することができるため、それによりサーバーの許可機構が使用不可になる恐れがあります。このためファイルを書き込み許可の制限されたディレクトリーに入れておく必要があります。</p>
DisplayManager.DISPLAY.authComplain	<p>このリソースに False (偽) を設定した場合、ログイン・ウィンドウの unsecureGreeting が使用不可になります。『認証クライアント』のセクションを参照してください。デフォルトは True (真) です。</p>
DisplayManager.DISPLAY.resetSignal	<p>xdm コマンドがサーバーをリセットするために送信するシグナルの数です。『サーバーの制御』のセクションを参照してください。デフォルトは 1 (SIGHUP) です。</p>
DisplayManager.DISPLAY.termSignal	<p>xdm コマンドがサーバーを終了させるために送信するシグナルの数です。『サーバーの制御』のセクションを参照してください。デフォルトは 15 (SIGTERM) です。</p>

項目	説明
<code>DisplayManager.DISPLAY.resetForAuth</code>	許可ファイルのセットアップ後、 <code>xdm</code> コマンドがサーバーに <code>SIGHUP</code> を送信するかどうかを指定します。 <code>SIGHUP</code> を送信すると、新しい権限情報の読み取り中に追加のサーバーがリセットされます。デフォルト値は <code>False</code> です。これは、すべての <code>AIXwindows</code> サーバーに有効です。
<code>DisplayManager.DISPLAY.userAuthDir</code>	<code>xdm</code> コマンドが通常のユーザー許可ファイル (<code>\$HOME/.Xauthority</code>) に書き込めなくなったとき、このディレクトリーに固有なファイルを作成し、 <code>XAUTHORITY</code> 環境変数にそのファイル名を設定します。デフォルトでは、 <code>/tmp</code> を使用します。

XDMCP アクセス制御

`xdm` コマンドは、`DisplayManager.accessFile` リソースで指定されたデータベース・ファイルが提供する情報を使って、`XDMCP` サービスを要求するディスプレイからのアクセスを制御します。このファイルには、下記の 3 つのエントリーが入っています。

- **Direct** および **Broadcast** 照会への応答を制御するエントリー
- **Indirect** 照会への応答を制御するエントリー
- マクロ定義

Direct 照会エントリーには、ホスト名またはパターンが入っています。パターンとは、ホスト名と区別するために、1 つ以上のパターン・マッチング文字が入った文字列のことです。* (アスタリスク) は 0 またはそれ以上の任意の文字と突き合わせし、? (疑問符) は任意の 1 文字と突き合わせします。エントリーはディスプレイ・デバイスのホスト名と比較されます。エントリーがホスト名の場合、ネットワーク・アドレスを使ってすべての比較が行われ、正しいネットワーク・アドレスに変換されたすべての名前が使用できます。エントリーがパターンの場合、実際のホスト名だけが比較に使用されるため、別名と一致させるようなことはしないでください。(感嘆符) の後にホスト名またはパターンを続けると、エントリーと一致するホストは比較から除外されます。

Indirect (間接) エントリーにも、ホスト名かパターンが入っていますが、その後ろに **Indirect** 照会の送信先であるホスト名やマクロのリストが続きます。

マクロ定義には、マクロ名と、ホスト名のリストとそのマクロが展開する他のマクロが入っています。ホスト名とマクロを区別するため、マクロ名の前に % (パーセント) 文字が付いています。マクロはネストできます。

Indirect (間接) エントリーには、`xdm` コマンドに `chooser` コマンドを実行させて、接続先のホストのメニューを提供するように指示することもできます。詳しくは、`Chooser` のセクションを参照してください。

特定のディスプレイ・ホストのアクセスを検査するとき、各エントリーが順番にスキャンされ、最初に一致したエントリーが応答を決定します。例えば、**Indirect** エントリーのスキャン時には **Direct** 照会エントリーは無視されます。**Direct** エントリーのスキャン時には **Broadcast** 照会エントリーは無視されます。

空白行は無視されます。# は、コメント区切り文字として扱われ、その行の残りは無視されます。行末に ¥ (円記号) を指定すると改行が無視され、間接ホストのリストを複数の行に渡って入力することができます。

下記に、`Xaccess` ファイルの例を示します。

```
#
# Xaccess - XDMCP access control file
#
```

```

#
# Direct/Broadcast query entries
#
!xtra.lcs.mit.edu      # disallow direct/broadcast service for xtra
bambi.ogi.edu        # allow access from this particular display
*.lcs.mit.edu        # allow access from any display in LCS
#
# Indirect query entries
#
%HOSTS                expo.lcs.mit.edu xenon.lcs.mit.edu ¥¥
                    excess.lcs.mit.edu kanga.lcs.mit.edu
extract.lcs.mit.edu   xenon.lcs.mit.edu #force extract to contact xenon
!xtra.lcs.mit.edu     dummy          #disallow indirect access
*.lcs.mit.edu        %HOSTS          #all others get to choose

```

Chooser

X 端末が **Broadcast** または **Indirect** 照会用のホスト・メニューを提供しない場合、**chooser** プログラムがこの役目を引き受けます。 **Xaccess** ファイル内で、**CHOOSER** を間接ホスト・リストの最初のエントリーに指定してください。 **chooser** プログラムは、リストの残りのホストそれぞれに **Query** 要求を送信し、それに応答するホストすべてにメニューを提供します。

また、このリストに **BROADCAST** という単語を指定すると、**chooser** は **Broadcast** 照会を送信し、再度、応答したホストすべてにメニューを提供します。

下記は、**chooser** を使った **Xaccess** ファイルの例です。

```

extract.lcs.mit.edu    CHOOSER %HOSTS      #offer a menu of these hosts
xtra.lcs.mit.edu      CHOOSER BROADCAST  #offer a menu of all hosts

```

chooser で使用されるプログラムは、**DisplayManager.DISPLAY.chooser** リソースで指定します。このプログラム用のリソースは、**DisplayManager.DISPLAY.resources** リソースで指定されたファイルに入れておくことができます。

chooser は、**Motif SelectionBoxWidget** を使用してインストールされています。リソース、ウィジェット名、またはガジェット名の詳細については、**XmSelectionBoxWidget Class** の資料を参照してください。

サーバー仕様

DisplayManager.servers というリソースは、サーバーの仕様を提供します。値が / (スラッシュ) で始まっている場合、サーバー仕様の入ったファイル名が 1 行に 1 つずつあります。

各仕様は、**XDMCP** を使わずに常に管理する必要のあるディスプレイを指定します。それぞれ、最低でも 3 つの部分からなります。

- 表示名
- ディスプレイのクラス
- ディスプレイのタイプ
- ローカル・サーバーの場合は、サーバーを始動するコマンド・ライン

ローカル・ディスプレイ番号 0 の一般的なエントリーは、下記ようになります。

```
:0 IBM-GT local /usr/bin/X11/X :0
```

ディスプレイ・タイプには、下記の 2 種類があります。

項目	説明
local	ローカル・ディスプレイ: <code>%f!xdm%fP</code> がサーバーを実行する必要があります。
foreign	リモート・ディスプレイ: <code>%f!xdm%fP</code> が実行中のサーバーへの X 接続をオープンします。

ディスプレイ名は、**-display** オプションで X プログラムに渡せるものである必要があります。この文字列はディスプレイ固有のリソース名を生成するのに使用されるため、名前が一致するように気を付けてください (例えば、ユーザーのリソースが "DisplayManager._0.session" の場合、"`localhost:0 local /usr/bin/X11/X :0`" ではなく、"`:0 local /usr/bin/X11/X :0`" を使用します)。ディスプレイ・クラスも、ディスプレイ特有のリソースの中でリソースのクラスとして使用されます。これは、同じようなディスプレイ (例えば、X 端末) が多くあり、リソースをグループ別にセットする場合に便利です。XDMCP を使用するとき、ディスプレイには、ディスプレイ・クラスを指定する必要があります。個々の X 端末の説明書には、ユーザーのデバイスに対するディスプレイ・クラスの文字列が記載されているはずですが、説明書に記載されていない場合は、**xdm** コマンドをデバッグ・モードで実行し、ユーザーのデバイスに対して生成されるリソース文字列を調べてください。この中に、ディスプレイ・クラスを表す文字列が入っています。

セットアップ・プログラム

Xsetup ファイルは、サーバーのリセット後、ログイン・ウィンドウが提供される前に実行されます。通常、このファイルはシェル・スクリプトです。これは **root** として実行されるため、セキュリティについては十分な注意が必要です。これは、ルートをバックグラウンドに変更するか、ログイン・ウィジェットとともに画面に表示される他のウィンドウを始動するためのものです。**xdm** はキーボードをGrabするため、他のウィンドウはキーボード入力を受け取ることができなくなります。しかし、マウスを使って対話することが可能であるため、ここでは潜在的なセキュリティ・ホールに注意する必要があります。

DisplayManager.DISPLAY.grabServer が設定されている場合、**Xsetup** はまったくディスプレイと接続することができません。このプログラムのリソースは、**DisplayManager.DISPLAY.resources** で指定されたファイルに入れておくこともできます。

DisplayManager.exportList で指定されたものに加えて、下記の環境変数が渡されます。

項目	説明
DISPLAY	関連するディスプレイ名を指定します。
PATH	DisplayManager.DISPLAY.systemPath の値を指定します。
SHELL	DisplayManager.DISPLAY.systemShell の値を指定します。
XAUTHORITY	許可ファイルに設定することを指定します。

認証クライアント

MIT 認証ウィジェットは、標準 Motif ウィジェットからなる認証クライアントで置き換えられています。下記のリストは、ウィジェット名 (およびそのウィジェット・クラス) を示しています。

```

outframe(xmFrameWidget)
  inframe(xmFrameWidget)
    main(XmFormWidget)
      tframe(xmFrameWidget)
        greeting(xmLabelGadget)
      logoLine(xmFormWidget)
        dpyname(xmLabelWidget)
      userLine(xmRowColumnWidget)
        userLabel(xmLabelWidget)
        username(xmTextWidget)
        passLabel(xmLabelWidget)
        password(xmTextWidget)
      failSafeLine(xmFormWidget)

```

```
failsafe(xmToggleButtonWidget)
cancelline(xmFormWidget)
cancel(xmPushButtonWidget)
message(xmLabelWidget)
```

認証クライアントは、キーボードから名前とパスワードの組を読み取ります。このクライアントのリソースは、**DisplayManager.DISPLAY.resources** という名前のファイルに入れてください。これらのリソースはすべて、適切なデフォルト値を持っているため、特に指定する必要はありません。認証クライアント・リソースのデフォルト値に関する詳細については、該当するウィジェット・クラスの資料、および **/usr/lib/X11/xdm/Xresources** を参照してください。下記のリソースも認証クライアントによってサポートされます。

項目	説明
Xlogin*foreground	フォアグラウンドに使用する色を指定します。
Xlogin*background	バックグラウンドに使用する色を指定します。
Xlogin*greeting	このウィンドウを識別する文字列を指定します。デフォルトは AIXwindows 環境 です。
Xlogin*greetFont	あいさつメッセージを表示するフォントを指定します。
Xlogin*frameColor	あいさつメッセージの表示に使用するバックグラウンドの色を指定します。
Xlogin*titleFont	タイトルの表示に使用するフォントを指定します。
Xlogin*namePrompt	ユーザー名の入力を求めるプロンプトを表示する文字列を指定します。Xrdb プログラムは、リソースの値の最後に付いているホワイト・スペースを取り除きます。プロンプトの最後にスペースを入れる場合は、逆スラッシュでスペースをエスケープしてください。デフォルトは「login」です。
Xlogin*promptFont	両者の入力を求めるプロンプトに使用するフォントを指定します。
Xlogin*failFont	フェイルセーフ・ボタンに使用するフォントを指定します。
Xlogin*cancelFont	キャンセル・ボタンに使用するフォントを指定します。
Xlogin*messageFontlist	失敗のメッセージを表示するのに使用するフォントを指定します。
Xlogin*failColor	失敗のメッセージの表示に使用する色を指定します。
Xlogin*failTimeout	失敗メッセージを表示する時間 (秒) を指定します。デフォルトは 30 秒です。
Xlogin*sessionArgument	セッション・プログラムに渡す引数を指定します。

項目

Xlogin*XmText.translations

説明

このリソースは、認証クライアントに使用する変換を指定します。変換の詳細については、X Toolkit の資料を参照してください。デフォルトの変換テーブルは下記のとおりです。

```
Ctrl<Key>b: backward-character() %n%
Ctrl<Key>a: beginning-of-line() %n%
Ctrl<Key>e: end-of-line() %n%
Ctrl<Key>f: forward-character() %n%
Ctrl<Key>d: kill-next-character() %n%
Ctrl<Key>k: kill-to-end-of-line() %n%
Ctrl<Key>u: kill-to-start-of-line() %n%
```

XDM をセットアップして、Xresources に定義されている XmText 変換を置き換え、標準の XDM 変換を使用することが可能です。

注: Motif のデフォルトの仮想割り当てのため、<Key>osfHelp を <Key>F1 の代わりに使用してください。

```
Xlogin*XmText.translations: #override %n%
```

```
<Key>osfHelp: set-session-argument(failsafe) finish-field() %n%
Ctrl<Key>Return: set-session-argument(failsafe) finish-field() %n%
Ctrl<Key>H: delete-previous-character() %n%
Ctrl<Key>D: delete-character() %n%
Ctrl<Key>B: move-backward-character() %n%
Ctrl<Key>F: move-forward-character() %n%
Ctrl<Key>A: move-to-beginning() %n%
Ctrl<Key>E: move-to-end() %n%
Ctrl<Key>K: erase-to-end-of-line() %n%
Ctrl<Key>U: erase-line() %n%
Ctrl<Key>X: erase-line() %n%
<Key>Return: finish-field() %n
<Key>BackSpace: delete-previous-character() %n%
<Key>Delete: delete-previous-character() %n
```

一般の XmText アクションの他に、標準 XDM 変換と互換性を保つために、クライアントは下記のアクションもサポートします。

delete-previous-character

カーソルの前の文字を消去します。

delete-character

カーソルの後の文字を消去します。

move-backward-character

カーソルを逆方向に移動します。

move-forward-character

カーソルを順方向に移動します。

move-to-beginning

カーソルを編集可能テキストの先頭に移動します。

move-to-end

カーソルを編集可能テキストの終わりに移動します。

erase-to-end-of-line

カーソルの後のテキストをすべて消去します。

erase-line

テキスト全体を消去します。

finish-field

カーソルが name フィールド内にあれば、password フィールドに移動します。カーソルが password フィールド内にあれば、現在の名前とパスワードの組を検査します。名前とパスワードの組が有効であれば、**xdm** はセッションを開始します。それ以外の場合は、失敗メッセージが表示され、プロンプトが再表示されます。

insert-char

入力された文字を挿入します。

set-session-argument

始動時にセッションに渡される単一の語の引数を指定します。セッション・プログラムおよび一般的な使用方法のセクションを参照してください。

始動プログラム

通常、**Xstartup** ファイルはシェル・スクリプトです。これは **root** として実行されるため、実行時のセキュリティには十分な注意が必要です。このファイルには、**/etc/utmp** へのエントリーの追加、ファイル・サーバーからのユーザーのホーム・ディレクトリーのマウント、日付メッセージの表示、ログイン失敗時のセッションのキャンセルを行うコマンドが入っています。

DisplayManager.exportList で指定された環境変数に加え、下記の変数が渡されます。

項目	説明
DISPLAY	関連するディスプレイ名を指定します。
HOME	ユーザーの初期作業ディレクトリーを指定します。
USER	ユーザー名を指定します。
PATH	DisplayManager.DISPLAY.systemPath の値を指定します。
SHELL	DisplayManager.DISPLAY.systemShell の値を指定します。
XAUTHORITY	許可ファイルに設定することを指定します。

スクリプトに渡される引数はありません。**xdm** コマンドはこのスクリプトが終了するまで待機し、それからユーザー・セッションが始まります。このスクリプトの終了値がゼロ以外の場合、**xdm** コマンドはセッションを中断し、他の認証サイクルを開始します。

セッション・プログラム

Xsession プログラムは、ユーザーのセッションのスタイルを確立します。このプログラムは、許可ユーザーが許可することで実行します。

DisplayManager.exportList で指定されたものに加えて、下記の環境変数が渡されます。

項目	説明
DISPLAY	関連するディスプレイ名を指定します。
HOME	ユーザーの初期作業ディレクトリーを指定します。
USER	ユーザー名を指定します。
PATH	DisplayManager.DISPLAY.userPath の値を指定します。
SHELL	ユーザーのデフォルトのシェルを (getpwnam から) 指定します。
XAUTHORITY	標準でない許可ファイルに設定することを指定します。

ほとんどの場合、**Xsession** プログラムはインストール時に、ユーザーのホーム・ディレクトリー (**\$HOME**) に **.xsession** というファイルがあるかどうか調べます (このファイルには、ユーザーがセッションとして使用したいコマンドが入っています)。ユーザーが指定したセッションが存在しなければ、**Xsession** プログラムがシステムのデフォルト・セッションとしてインストールされます。一般的な使用方法のセクションを参照してください。

'set-session-argument' アクションを使って、認証ウィジェットからこのプログラムに引数を渡すことができます。これは、異なるスタイルのセッションを選択するときに使用します。この機能を使うと、通常のセッションに失敗したときに、ユーザーはこのセッションからエスケープすることができます。したがって、ユーザーは失敗したとき、システム管理者に照会しなくても自分の **.xsession** を修理することができます。一般的な使用方法のセクションには、この機能についての説明があります。

リセット・プログラム

Xreset スクリプトは、ユーザーのセッションが終了したときに実行されます。このスクリプトは **root** として実行されるため、**Xstartup** スクリプトの中で指定されたコマンドの効力を取り消すコマンド (**/etc/utmp** からエントリーを除去したり、ファイル・サーバーからディレクトリーをアンマウントしたり

する)が入っている必要があります。 **Xstartup** に渡される環境変数は、**Xreset** にも渡されます。このプログラムは、**Xstartup** プログラムとシンメトリックです。

サーバーの制御

xdm コマンドは、POSIX シグナルを使ってローカル・サーバーを制御します。 **SIGHUP** シグナルは、サーバーをリセットするとき、すべてのクライアント接続をクローズするとき、および他のクリーンアップ作業を行うときに使用されます。 **SIGTERM** シグナルは、サーバーをキャンセルするときに使用されます。これらのシグナルが期待されたアクションを行わない場合、**DisplayManager.DISPLAY.resetSignal** および **DisplayManager.DISPLAY.termSignal** リソースで代替シグナルを指定することができます。

XDMCP を使用しないリモート端末を制御する場合、**xdm** コマンドはディスプレイ上のウィンドウ階層を検索し、**KillClient** プロトコル要求を使って次のセッションのために端末のクリーンアップを試みます。実際には、すべてのクライアントが非アクティブにならないことがあります。これはウィンドウを作成したクライアントだけが通知されるためです。 **XDMCP** は、より確実な機構を提供します。つまり、**xdm** コマンドが初期接続をクローズすると、セッションは終了し、端末は他のすべての接続をクローズするよう要求されます。

XDM の制御

xdm コマンドは、2つのシグナル **SIGHUP** と **SIGTERM** に応答します。 **SIGHUP** が送信されると、**xdm** は再度構成ファイル、アクセス制御ファイル、およびサーバー・ファイルを読み取ります。サーバー・ファイルの場合、エントリーが追加または除去されたかどうかを通知します。新しいエントリーが追加されていた場合、**xdm** コマンドは関連するディスプレイ上でセッションを開始します。エントリーが除去されていた場合、そのエントリーは即座に使用不可になります。つまり、進行中のセッションは通知なしに終了され、新しいセッションは開始されません。

SIGTERM が送信されると、**xdm** コマンドは進行中のすべてのセッションを停止し終了します。これは、システムをシャットダウンするときに使用します。

xdm コマンドは、コマンド・ライン引数リストを適切に編集して、**ps** コマンドで使用するさまざまなサブプロセスにマークを付けようとします。**xdm** コマンドはこの作業のために余分なスペースを割り当てることができないため、妥当な長さのコマンド・ラインで **xdm** コマンドを始動してください (絶対パス名で十分です)。ディスプレイにリソースを提供する各プロセスには、**-display** というマークが付けられます。

その他の機能

xdm コマンドでは、**xinit** コマンド・オプションまたは他の適切なデーモンを使用してコマンド・ラインにサーバーを指定すると一度に単一のセッションを実行することができます。

```
xdm -server ":0 local /usr/bin/X11/X :0 -force"
```

また、ファイル・サーバーと複数の X 端末を実行することもできます。この場合の構成は、**Xservers** ファイルが下記のような点を除いて、前の例と同じです。

```
extol:0 VISUAL-19 foreign
exalt:0 NCD-19 foreign
explode:0 NCR-TOWERVIEW3000 foreign
```

これにより **xdm** コマンドが上記 3つの端末すべてのセッションを管理します。 **XDM** の制御のセクションには、これらの端末の使用をシグナルで制御する方法についての説明があります。

注: **xdm** コマンドは、他のウィンドウ・システムとうまく共存することができません。複数のウィンドウ・システムを同一ハードウェアで使用する場合は、**xinit** コマンドを使用してください。

例

1. 下記の **xstartup** スクリプトは、**/etc/nologin** ファイルが存在している間はログインを禁止する例です。ここでは何らかのメッセージを表示する用意がなく (ファイルを表示するコア X クライアントがない)、説明のないままログインが失敗する可能性があるため、この例のセットアップはお勧めしません。つまり、これは完全な例ではなく、単に使用可能な機能をデモンストレーションするためのものです。

```
#!/bin/sh
#
# Xstartup
#
# This program is run as root after the user is verified
#
if [ !-f /etc/nologin ]; then
    exit 1
fi
exit 0
```

2. 下記の **Xsession** スクリプトは、通常のセッションから抜け出すために上記 **Xresources** ファイルの変換に指定された特殊な **failsafe** モードを認識します。

```
#!/bin/sh
exec > $HOME/.xsession-errors 2>&1
case $# in
1)
    case $1 in failsafe)
        exec aixterm -geometry 80x24-0-0
        ;;
    esac
esac
startup=$HOME/.xsession
resources=$HOME/.Xresources
if [ -f /usr/bin/X11/startx ]; then
    exec /usr/bin/X11/startx -t -wait
elif [ -f $startup ]; then
    exec $startup
else
    if [ -f $resources ]; then
        xrdb -load $resources
    fi
    mwm &
    exec aixterm -geometry 80x24+10+10 -ls
fi
```

3. **xdm** をシステム始動時に使用可能にするには、**root** として下記のように入力します。

```
/usr/lib/X11/xdm/xdmconf
```

4. **xdm** をリブート時に使用不可にするには、**root** として下記のように入力します。

```
/usr/lib/X11/xdm/xdmconf -d
```

5. **xdm** を使用してディスプレイを管理する際、認証プロシージャーによってディスプレイに接続できるのは許可されているクライアントのみであることが保証されます。X11 R4 および X11 R5 ライブラリーを使用して構築されたクライアントでは、このプロトコルが認識されます。X11 R3 以前のライブラリーを使用して構築されたクライアントでは、この認証プロトコルがサポートされないため、**xhost** 許可を取得しなければ X サーバーには接続できません。下記のように入力して、ローカル・クライアントへ接続することができます。

```
xhost =localhost
```

または

```
xhost =machine
```

この場合、*machine* はローカル・クライアントのホスト名です。

ファイル

項目	説明
<code>/usr/lib/X11/xdm/xdm-config</code>	デフォルトの構成ファイル。
<code>/usr/lib/X11/xdm/Xaccess</code>	デフォルトのアクセス・ファイル。許可されたディスプレイがリストされています。
<code>/usr/lib/X11/xdm/Xservers</code>	デフォルトのサーバー・ファイル。管理すべき非 XDMCP サーバーがリストされています。
<code>\$(HOME)/.Xauthority</code>	ユーザー許可ファイル。 <code>xdm</code> は、クライアントが読み取るべきキーをこのファイルに保管します。
<code>/usr/lib/X11/xdm/chooser</code>	デフォルトの <code>chooser</code> 。
<code>/usr/bin/X11/xrdb</code>	デフォルトのリソース・データベース・ローダー。
<code>/usr/bin/X11/X</code>	デフォルトのサーバー。
<code>/usr/bin/X11/xterm</code>	デフォルトのセッション・プログラムとフェイルセーフ・クライアント。
<code>/usr/lib/X11/xdm/A<host>¥-<suffix></code>	許可ファイルのデフォルト位置。

関連資料:

143 ページの『X コマンド』

190 ページの『xinit コマンド』

関連情報:

startx コマンド

xfindproxy コマンド

目的

プロキシ・サービスを見つけます。

構文

```
xfindproxy -manager ManagerAddress -name ServiceName -server ServerAddress [ -auth ] [ -host HostAddress ] [ -options Options ]
```

説明

`xfindproxy` は、使用可能なプロキシ・サービスを見付けるために使用するプログラムで、プロキシ管理プロトコルを使用して、プロキシ・マネージャーと通信します。プロキシ・マネージャーは使用可能なすべてのプロキシ・サービスを追跡し、必要であれば新規のプロキシを始動し、可能であればいつでもプロキシが共用されるようにします。

`xfindproxy` が正常にプロキシ・アドレスを取得した場合は、`stdout` に出力されます。プロキシ・アドレスは、使用されるプロキシ・サービスに固有の形式を持っています。例えば、LBX のプロキシ・サービスの場合、プロキシの X ディスプレイ・アドレス (例 `blah.x.org:63`) がプロキシ・アドレスになります。

`xfindproxy` が正常にプロキシ・アドレスを取得できなかった場合は、エラーが `stderr` に出力されます。

フラグ

項目	説明
-manager	この引数は必須で、プロキシ・マネージャーのネットワーク・アドレスを指定します。アドレスの形式は、標準 ICE ネットワーク ID (例えば、tcp/blah.x.org:6500) です。
-name	この引数は必須で、希望するプロキシ・サービスの名前 (例えば、LBX) を指定します。名前は、大/小文字を区別しません。
-server	この引数も必須で、ターゲット・サーバーのアドレスを指定します。アドレスは、 -name 引数で指定されたプロキシ・サービスに固有の形式を持っています。例えば、LBX のプロキシ・サービスの場合、X ディスプレイ・アドレス (例 blah.x.org:0) がアドレスになります。
-auth	この引数はオプションです。指定された場合、 xfindproxy は、標準入力から 2 行読み取ります。最初の行は、許可/認証名です。2 番目の行は、16 進形式 (xauth で使用されるものと同じ形式) での許可/認証データです。 xfindproxy は、この auth データをプロキシに渡し、ほとんどの場合、それ自体をターゲット・サーバーに対して許可/認証するために、プロキシが使用します。
-host	この引数はオプションです。 xfindproxy が新規のプロキシ・サービスを始動すると、そのサービスは指定のホストに渡されます。プロキシは、このホストに対するすべての接続の制限を選択することができます。 xfindproxy が既存のプロキシを見付けた場合は、ホストはパスされますが、プロキシがこのホストを使用する方法についてのセマンティクスは未定義です。
-options	この引数はオプションです。 xfindproxy が新規のプロキシ・サービスを始動すると、そのサービスは指定された任意のオプションを渡します。オプションのセマンティクスは、各プロキシ・サーバーに固有で、ここでは定義されません。 xfindproxy が既存のプロキシを見付けた場合は、オプションはパスされますが、プロキシがこのオプションを使用する方法についてのセマンティクスは未定義です。

関連情報:

proxymngr コマンド

xfs コマンド

目的

X Window System のディスプレイ・サーバーにフォントを提供します。

構文

```
xfs [ -config ConfigurationFile ] [ -ls ListenSocket ] [ -port Number ]
```

説明

xfs は、AIXwindows のフォント・サーバーです。AIXwindows のディスプレイ・サーバーにフォントを提供します。

xfs サーバーは、下記のシグナルに応答します。

項目	説明
SIGTERM	フォント・サーバーが、すべて終了します。
SIGUSR1	サーバーが、構成ファイルを再読み取りします。
SIGUSR2	サーバーが、保持しているキャッシュ・データをフラッシュします。
SIGHUP	サーバーが、すべてのアクティブな接続をクローズし、構成ファイルを再読み取りし、リセットします。

サーバーは通常、システム管理者によって実行され、`/etc/rc.tcpip` などのブート・ファイル経由で始動します。ユーザーも、特定のフォント・セット用の専用フォント・サーバーを始動することができます。

構成言語は、キーワードと値の組のリストです。各キーワードの後には、`=` (等号) および指定したい値を続けます。

次のリストに、認識されるキーワードと、および有効な値の型と説明を示します。

項目	説明
#	1 桁目に位置する場合には、コメント文字。
catalogue (文字列のリスト)	フォント・パス・エレメント名の順序付けられたリスト。現在のインプリメントでは、指定されたすべてのフォントが含まれる、単一のカタログ ("すべて") のみをサポートしています。
alternate-servers (文字列のリスト)	当該フォント・サーバーの代替サーバーのリスト。
client-limit (基数)	当該フォント・サーバーがサービスを拒否する前にサポートするクライアントの数。これは、個別の各フォント・サーバーに対してロードをチューニングする際に便利です。
clone-self (ブール)	client-limit に達したとき、このフォント・サーバーがサーバー自体のクローンを作成すべきかどうかを指定します。
default-point-size (基数)	サイズが指定されていないフォントに対するデフォルトのポイント・サイズを (デシポイントで) 指定します。
default-resolutions (レゾリューションのリスト)	サーバーがデフォルトでサポートするレゾリューション。この情報は、事前レンダリングを行う際の手掛かりとなり、レゾリューションの指定されていないスケール・フォントを置き換えます。
error-file (文字列)	レゾリューションはコンマで区切られた、1 インチ当たりのピクセル数で示される x と y の組です。複数のレゾリューションは、コンマで区切ります。 エラー・ファイルのファイル名。すべての警告とエラーは、ここに記録されます。
port (基数)	サーバーが接続を <code>listen</code> する TCP ポート。デフォルトは 7100 です。
use-syslog (ブール)	syslog 関数 (サポートされているシステム上に存在) をエラー時に使用するかどうかを指定します。
deferglyphs (文字列)	遅延フェッチと <code>glyphs</code> のキャッシュに対するモードを設定します。値 <code>none</code> は、遅延の <code>glyphs</code> が使用不可であることを意味します。 <code>all</code> は、遅延の <code>glyphs</code> がすべてのフォントに対して使用可能であることを意味します。また <code>16</code> は、遅延の <code>glyphs</code> が 16 ビットのフォントに対してのみ使用可能であることを意味します。

下記のフォーマットの 1 つは、TCP 接続を受け入れるフォント・サーバーに名前を付けるために使用されるものです。

```
tcp/hostname:port
tcp/hostname:port/cataloguelist
```

`hostname` は、フォント・サーバーが実行中であるマシンの名前 (あるいは 10 進数アドレス) を指定します。 `port` は、フォント・サーバーが接続を `listen` している 10 進数の TCP ポートです。 `cataloguelist` は、セパレーターとしての '+' を含むカタログ名のリストです。以下に例をいくつか示します。

```
tcp/expo.lcs.mit.edu:7100, tcp/18.30.0.212:7101/all
```

下記のフォーマットの 1 つは、DECnet 接続を受け入れるフォント・サーバーに名前を付けるために使用されるものです。

```
decnet/nodename::font$objname  
decnet/nodename::font$objname/catalogelist
```

`nodename` は、フォント・サーバーが実行中であるマシンの名前 (あるいは 10 進数アドレス) を指定します。`objname` は、正規の、大/小文字を区別しない、DECnet のオブジェクト名です。`catalogelist` は、セパレーターとしての '+' を含むカタログ名のリストです。

フラグ

項目	説明
-config <i>ConfigurationFile</i>	フォント・サーバーが使用する構成ファイルを指定します。
-ls <i>ListenSocket</i>	listen ソケットとして使用するために既にセットアップされた、ファイル・ディスクリプターを指定します。このオプションは、フォント・サーバー自体が自動的にサーバー自体のコピーを <code>spawn</code> して追加の接続を操作するときのみ、使用されるものです。
-port <i>Number</i>	サーバーが接続を <code>listen</code> する TCP ポート番号を指定します。

例

```
#  
# sample font server configuration file  
#  
# allow a max of 10 clients to connect to this font server  
client-limit = 10  
# when a font server reaches its limit, start up a new one  
clone-self = on  
# alternate font servers for clients to use  
alternate-servers = hansen:7101,hansen:7102  
# where to look for fonts  
# the first is a set of Speedo outlines, the second is a set of  
# misc bitmaps and the last is a set of 100dpi bitmaps  
#  
catalogue = /usr/lib/fonts/type1,  
            /usr/lib/X11/ncd/fonts/misc,  
            /usr/lib/X11/ncd/fonts/100dpi/  
# in 12 points, decipoints  
default-point-size = 120  
# 100 x 100 and 75 x 75  
default-resolutions = 100,100,75,75
```

ファイル

項目
/usr/lib/X11/fs/config

説明
デフォルトの構成ファイル。

xget コマンド

目的

セキュア通信チャンネル内で秘密メールを受信します。

構文

xget

説明

xget コマンドは、セキュア通信チャンネル内で秘密メールを受信するために使用します。意図した受信者だけがメッセージを読み取れます。**xget** コマンドはユーザーにパスワードを要求して、秘密メールを読み取れるようにします。

秘密メールを送受信するには、**xget** コマンドを、**enroll** コマンドおよび **xsend** コマンドと一緒に使用します。**enroll** コマンドを使用してパスワードを設定し、秘密メールを受信します。**xsend** コマンドは、意図した受信者だけが読み取れるメールを送信します。

xget コマンドを発行すると、暗号キーの入力を求めるプロンプトが表示されます。**enroll** コマンドを使用して以前に設定したパスワードを入力します。

xget コマンドのプロンプトは ? (疑問符) です。下記のサブコマンドがメッセージの後処理を制御します。

項目

q (終了)

n (削除) または **d** (削除) または **Enter**

!*Command*

s[*Filename*]

w[*Filename*]

? (ヘルプ)

説明

まだ削除されていないメールをユーザーのメールボックスに書き込み、終了します。ファイル終わり (Ctrl-D) を押した場合と同じ効果があります。

現行メッセージを削除し、次のメッセージを表示します。

指定されたワークステーションのコマンドを実行します。

デフォルト・メール・ファイル **mbox** 内ではなく、指定された *File* パラメーター内にメッセージを保管します。

デフォルト・メール・ファイル **mbox** 内ではなく、指定された *File* パラメーター内に、ヘッダーを付けずにメッセージを保管します。

サブコマンドの要約を表示します。

例

1. 秘密メールを受け取るには、下記のように入力します。

```
xget
```

enroll コマンドとともに実行されると、パスワードの入力を求めるプロンプトが表示されます。パスワードを入力すると、**xget** コマンド・プロンプト (?) が表示され、秘密メールのリストが表示されます。

2. **xget** プロンプト (?) で、秘密メールを表示するためには、**Enter** キーを押します。

最新のメッセージが表示された後、? (疑問符) が表示されて **xget** コマンドがいずれかの **xget** サブコマンドを待っていることを示します。 **help** または (疑問符) を入力して使用可能なサブコマンドをリストします。

3. メッセージまたはファイルをデフォルトのメール・ファイルに保管するには、下記のように入力します。

```
xget
```

(疑問符) プロンプトが表示された後、必要なファイルが表示されるまで **Enter** キーを押します。該当するファイルが表示されたら下記のように入力します。

```
s
```

この例では、ファイルはデフォルト・メール・ファイル **mbox** 内に保管されます。

4. メッセージまたはファイルを特定のファイルに保管するには、下記のように入力します。

```
xget
```

(疑問符) プロンプトが表示された後、必要なファイルが表示されるまで **Enter** キーを押します。該当するファイルが表示されたら下記のように入力します。

```
s mycopy
```

この例では、ファイルはデフォルト・メール・ファイルにではなく **mycopy** というファイルに保管されます。

5. メッセージを削除するには、下記のように入力します。

```
xget
```

(疑問符) プロンプトが表示された後、必要なファイルが表示されるまで **Enter** キーを押します。該当するファイルが表示されたら下記のように入力します。

```
d
```

この例では、現行ファイルが削除されます。

ファイル

項目	説明
<code>/var/spool/secretmail/User.key</code>	<i>User</i> 用の暗号化された鍵が入っています。
<code>/var/spool/secretmail/User.[0-9]</code>	<i>User</i> 用の暗号化されたメール・メッセージが入っています。
<code>/usr/bin/xget</code>	実行可能ファイルが入っています。

関連資料:

234 ページの『**xsend** コマンド』

関連情報:

enroll コマンド

mail コマンド

メールの管理

xhost コマンド

目的

現在のホスト・マシン上で拡張 X Window システム にアクセスしているユーザーを制御します。

構文

```
xhost [ + | - ] [ Name ]
```

説明

xhost コマンドは、X サーバーが接続を受け入れるコンピューターのリストに、ホスト名を追加、削除します。

このコマンドはディスプレイが接続されているコンピューター上で実行しなければなりません。 *-Host* パラメーターを使うと、アクセス・リストから名前を除去することができます。アクセス・リストから現在の名前を除去しないようにしてください。除去してしまった場合は、修正する前にシステムをログアウトしてください。

変数を付けずに **xhost** コマンドを入力すると、現在、使用されている X サーバーへのアクセスを認められたホストの名前とアクセス可能かどうかを示すメッセージが表示されます。

セキュリティのため、アクセス制御を左右するオプションは制御ホスト からしか実行できません。ワークステーションでは、サーバーと同じコンピューターになります。X 端末では、ログイン・ホストになります。

デフォルトでリモート名を使用可能にするために、リモート名を */etc/X?.hosts* ファイルに定義することができます。ここで、*?* はアクセスが可能になるディスプレイ番号です。

例えば、ディスプレイ *jeanne:0* は、*jeanne* というデフォルトのホスト名を使用するシステム上の */etc/X0.hosts* ファイル内で定義されたシステムからアクセスすることができます。ディスプレイ名およびファイル名の中の *0* は、定義済みのリモート・システムが拡張 X Window システム を通してアクセスできるディスプレイ番号を示しています。

フラグ

項目	説明
+Name	X サーバー・アクセス・リストに追加するホスト名を定義します (プラス記号の指定は任意です)。
- Name	X サーバー・アクセス・リストから除去するホスト名を定義します。既存の接続は切断されませんが、新規接続の試みは拒否されます。現在のコンピューターは除去できますが、その後の接続 (その後再び追加する試みも含めて) はできません。ローカル接続を復活するには、サーバーをリセットする方法しかありません (つまり、すべての接続を切断しなければなりません)。
+	アクセスに制限がないことを指定します。アクセス制御がオフになります。
-	アクセス制御がオンになります。

完全な *Name* は、以下に示す *family:name* の構文になっています。

inet インターネット・ホスト

local 空文字列である、1 つの名前のみを含みます。

注: *family* は、大/小文字の区別が行われます。名前のフォーマットは、*family* によって異なります。

xinit コマンド

目的

X Window System を初期化します。

構文

```
xinit [ [ Client ] Options ] [ - - [ Server ] [ Display ] Options ]
```

説明

xinit コマンドは、**/etc/init** から、あるいは複数のウィンドウ・システムを使用する環境で、X を直接始動できないようなシステム上の最初のクライアント・プログラム、および AIXwindows サーバーを始動するときに使用されます。最初のクライアントが終了すると、**xinit** コマンドは X サーバーを停止し終了します。

コマンド・ラインに特定のクライアント・プログラムが指定されていない場合、**xinit** コマンドはクライアント・プログラムを始動するためのファイルを探します。**xinit** コマンドは、**\$XINITRC** 環境変数を探します。ファイルが見つからなければ、このコマンドは次に **\$HOME/.xinitrc** ファイルを探します。それでもファイルが見つからなければ、このコマンドは下記のステップを実行します。

1. **xinit** コマンドは、次に **/usr/lib/X11/\$LANG/xinitrc** を探します。
2. 次に、**/usr/lpp/X11/defaults/\$LANG/xinitrc** を探します。
3. 最後に、**/usr/lpp/X11/defaults/xinitrc** を探します。

上記のファイルがすべて見つからなかった場合、**xinit** はデフォルトで次のコマンドを使用します。

```
aixterm ¥-geometry +1+1 ¥-n login ¥-display :0
```

コマンド・ラインに特定のサーバー・プログラムが指定されていない場合、**xinit** コマンドは下記のステップを実行します。

1. **xinit** コマンドは、サーバーを始動するためのシェル・スクリプトとして実行するためのファイルを探します。**xinit** コマンドはまず、**\$XSERVERRC** 環境変数の中のファイルを探します。
2. そのファイルがそこになれば、**\$HOME/.xserverrc** ファイルを探します。
3. **\$HOME/.xserverrc** ファイルが見つからなければ、次に、**/usr/lpp/X11/defaults/xserverrc** ファイルを探します。
4. 最後に、上記のファイルがすべて見つからなければ、**xinit** コマンドは、**X** コマンドを実行して X サーバーを始動し、デフォルトで次のコマンドを使用します。

```
X :0
```

これは現在の検索パスに X という名前のプログラムがあることを前提としている点に注意してください。しかし、サーバーは通常 **Xdisplaytype** と命名されます。ここで、**displaytype** とはこのサーバーによって使用可能になるグラフィック・ディスプレイのタイプを指します。したがって、サイトのシステム管理者は、コンピューターに適切なサーバーのタイプをリンクするか、あるいは、適切なサーバーで **xinit** コマンドを実行させるシェル・スクリプトを作成しなければなりません。

注: 使用可能なポインター・デバイス (マウスやタブレット) を使わないで AIXwindows を始動しようとしても、AIXwindows はオープンしません。デバイスを接続していても定義していなければ、システムはそのデバイスを認識しません。その逆も同様です。

重要な点は、**.xinitrc** で実行するプログラムは、すぐに終了しないのであればバックグラウンドで実行し、他のプログラムを始動するのを妨げないようにしなければなりません。しかし、寿命の長い最後に始動されたプログラム (通常はウィンドウ・マネージャーあるいは端末エミュレーター) はフォアグラウンドに残して、(ユーザーの終了および **xinit** の終了を指示する) スクリプトを終了させないようにします。

代替クライアントおよびサーバーはコマンド・ラインで指定することができます。指定したいクライアント・プログラムおよびその引数は、**xinit** に対して最初のコマンド・ラインの引数として設定されなければなりません。特定のサーバー・コマンド・ラインを指定するには、指定したいサーバー・コマンドが続く (クライアントおよび引数の後の) **xinit** コマンド・ラインに **—** (ハイフン 2 つ) を付けます。

クライアント・プログラム名およびサーバー・プログラム名は、**/** (スラッシュ) あるいは **.** (ピリオド) で始めなければなりません。そうでなければ、それらの名前はそれぞれの始動行に追加される引数として処理されます。これにより、すべてのコマンド・ラインを再入力しないで引数 (例えば、フォアグラウンド・カラーおよびバックグラウンド・カラー) を追加できます。

明確なサーバー名が指定されず、**—** (ハイフン 2 つ) の後に続く最初の引数が **:** (コロン) の後に続く数字である場合には、**xinit** は、ゼロの代わりにその数字をディスプレイ番号として使用します。残りのすべての引数はサーバーのコマンド・ラインに追加されます。

下記の環境変数が **xinit** コマンドで使用されます。

項目	説明
DISPLAY	この変数は、クライアントが接続するディスプレイの名前に設定します。
XINITRC	この変数は、初期ウィンドウを始動するシェル・コマンドを含む init ファイルを指定します。デフォルトではホーム・ディレクトリ内の .xinitrc が使用されます。
オプション	指定したクライアントで使用できるオプションのリストです。
<i>Client</i>	ユーザーが作業するクライアントを指定します。例えば、 xterm や aixterm です。指定するクライアントは、 . (ドット) か / (スラッシュ) で始まらなければなりません。
<i>Server</i>	有効な X サーバーを指定します。指定するサーバーは、 . (ドット) か / (スラッシュ) で始まらなければなりません。

例

1. X という名前のサーバーを始動し、ユーザーの **xinitrc** プログラムがあればそれを実行するか、または **aixterm** コマンドを始動するには、次のように入力します。

```
xinit
```

2. 代替ディスプレイ上で特定のタイプのサーバーを始動するには、下記のように入力します。

```
xinit -- /usr/bin/X11/X qdss:1
```

3. X という名前のサーバーを始動し、デフォルトの **xinitrc** コマンドまたは **aixterm** コマンドに引数を指定するには、下記のように入力します。

```
xinit -geometry =80x65+10+10 -fn 8x13 -j -fg white -bg navy
```

4. コマンド **/Xsun -l -c** を使用してサーバーを始動し、デフォルトの **xinitrc** コマンドまたは **aixterm** コマンドに引数 **-e widgets** を追加するには、下記のように入力します。

```
xinit -e widgets -- ./Xsun -l -c
```

5. 引数 **-a 2 -t 5** でディスプレイ 1 の X という名前のサーバーを始動し、ローカル・ワークステーションに表示を戻すコマンド **cpupig** を実行するマシン **fasthost** のリモート・シェルを始動するには、下記のように入力します。

```
xinit /usr/ucb/rsh fasthost cpupig -display ws:1 -- :1 -a 2 -t 5
```

6. 以下のサンプルの **.xinitrc** スクリプトは、クロックと複数の端末を始動させ、ウィンドウ・マネージャーを最終アプリケーションとして実行させておきます。ウィンドウ・マネージャーが適正に構成されているという前提で、ユーザーは「Exit (終了)」メニュー項目を選んで AIXwindows セッションを終了します。

```
xrdb -load $HOME/.Xresources
xsetroot -solid gray &
xclock -g 50x50-0+0 -bw 0 &
xload -g 50x50-50+0 -bw 0 &
xterm -g 80x24+0+0 &
xterm -g 80x24+0-0 &
mwm
```

7. 共通の始動環境を作成したいサイトが、サイト共通の始動ファイルを参照するデフォルトの **.xinitrc** スクリプトを簡単に作成できる例を下記に示します。

```
#!/bin/sh . /usr/local/lib/site.xinitrc
```

8. もう 1 つの方法は、特定のシェル・スクリプトで **xinit** コマンドを始動するスクリプトを記述することです。このようなスクリプトは、通常 **x11**、**xstart**、あるいは **startx** と命名され、初心者には簡単なインターフェースを提供する便利なものです。

```
#!/bin/sh xinit /usr/local/lib/site.xinitrc -- /usr/bin/X11/X bc
```

ファイル

項目	説明
.xinitrc	デフォルトのクライアント・スクリプト・ファイルが入っています。
 aixterm	.xinitrc がない場合にクライアントが実行するコマンドが入っています。
.xserverrc	デフォルトのサーバー・スクリプトが入っています。
X	.xserverrc がない場合にサーバーが実行するコマンドが入っています。

関連資料:

143 ページの『X コマンド』

関連情報:

startx コマンド

xkbcomp コマンド

目的

XKB キーボード記述をコンパイルします。

構文

```
xkbcomp [ -a ] [ -C ] [ -dflts ] [ -I Directory ] [ -l ] [ -m Name ] [ -merge ] [ -o OutputFile ] [ -opt Parts ] [ -R Directory ] [ -synch ] [ -w Level ] [ -xkb ] [ -xkm ] Source [ Destination ]
```

説明

xkbcomp コマンドは、XKB キーマップをいくつかの出力フォーマットの 1 つに変換する、キーマップ・コンパイラーです。 **xkbcomp** の最も一般的な使用法は、XKB 使用可能な X サーバーまたはユーティリティーによって直接読み取ることでできるコンパイル済みキーマップ・ファイル (拡張子 **.xkm**) を作成することです。このキーマップ・コンパイラーによって、C ヘッダー・ファイルまたは XKB ソース・ファイルも作成できます。 **xkbcomp** によって生成される C ヘッダー・ファイルは、組み込みデフォルト・キーマップを必要とする X サーバーまたはユーティリティーによってインクルードできます。

xkbcomp によって生成される XKB ソース・ファイルは完全に解決されており、XKB キーマップを通常構成するファイルが正しく組み合わされているかを確認するため、またはキーマップの完全な記述の入った単一のファイルを作成するために使用できます。

Source は X ディスプレイか、**.xkb** または **.xkm** ファイルを指します。*destination* のフォーマットは、明示的に指定されない限り、ソースのフォーマットによって決まります。**.xkb** (キーマップ・ソース) ファイルをコンパイルすると、デフォルトでは **.xkm** (コンパイルされたキーマップ・ファイル) が生成されます。ソースが **.xkm** ファイルまたは X ディスプレイの場合、**xkbcomp** はデフォルトでキーマップ・ソース・ファイルを生成します。

Destination が X ディスプレイの場合、ディスプレイのキーマップはコンパイルされたキーマップによって更新されます。

destination の名前は、通常はソースの名前から算出され、適切な拡張子が付けられます。いくつかのマップが入ったファイルから単一のマップをコンパイルする場合、**xkbcomp** は使用されるマップの名前に適切な拡張子が付けることによって、宛先ファイル名を構成します。

フラグ

項目	説明
-a	キーボード情報をすべて表示し、暗黙の情報または派生した情報をコメントとして示します。 .xkb フォーマット出力にのみ影響します。
-C	出力として C ヘッダー・ファイル (拡張子 .h) を生成します。
-dflts	存在していないコンポーネント (キー名など) のデフォルトを算出します。
-I Directory	キーマップ記述によってインクルードされるファイルを探す、トップレベルのディレクトリーを指定します。
-l	コマンド・ラインにリストされるファイルのマップ・パターンを指定するマップをリストします。
-m Name	複数の項目によってファイルからコンパイルされるマップを指定します。
-merge	コンパイルされた情報をサーバーからのマップとマージします。
-o Name	生成される出力ファイルの名前を指定します。デフォルトは、ソース・ファイルの名前に、出力フォーマットに応じて適切な拡張子を付けたものです。
-opt Parts	オプション・パーツを指定します。オプション・パーツのコンパイル・エラーは致命的なエラーではありません。 <i>Parts</i> は、文字 c 、 g 、 k 、 s 、 t の組み合わせで構成されます。 c は互換性マップ、 g は形状、 k はキー・コード、 s はシンボル、および t はタイプを指定します。
-R Directory	相対パス名のルート・ディレクトリーを指定します。
-synch	X 要求を強制的に同期します。
-w Level	コンパイル中の警告レポートを制御します。警告レベルを 0 にするとすべての警告が無効になります。警告レベルを 10 にするとすべての警告が有効になります。
-xkb	キーボードのソース記述を出力 (拡張子 .xkb) として生成します。
-xkm	コンパイルされたキーマップ・ファイルを出力 (拡張子 .xkm) として生成します。

xkbevd デーモン

目的

XKB イベント・デーモンです。

構文

```
xkbevd [ -help ] [ -cfg File ] [ -sc Command ] [ -sd Directory ] [ -display Display ] [ -bg ] [ -synch ] [ -v ]
```

説明

xkbevd イベント・デーモンは指定された XKB イベントを **listen** して、それが発生すると要求されたコマンドを実行します。構成ファイルには、イベントの指定/アクションの対および (または) 変数の定義が入っています。

イベントの指定は、短い XKB イベント名と、それに続けて修飾子として機能する文字列または ID を括弧に入れて指定します。括弧の中身が空の場合、修飾がないことを指定し、他の指定に一致しないイベントに適用されるデフォルト・コマンドを指定します。修飾子の解釈はイベントのタイプによって決まります。

- ベル・イベントはベルの名前を使用して突き合わせを行います。
- メッセージ・イベントは、メッセージ文字列の中身で突き合わせを行います。
- スロー・キー・イベントは、**press**、**release**、**accept**、または **reject** を受け入れます。

他のイベントは認識されません。

アクションは、オプション・キーワードと、それに続けてオプション文字列引数を指定します。**xkbevd** は以下に示すアクションを認識します。

- **none**
- **ignore**
- **echo**
- **printEvent**
- **sound**
- **shell**

アクションが指定されていない場合、文字列は、感嘆符で始まっていなければ、再生されるサウンド・ファイルの名前として取られます。感嘆符で始まっている場合、シェル・コマンドとして取られます。

引数文字列の変数定義は、引数文字列がアクション・プロセッサに渡される前に、当該のイベントからのフィールドによって拡張されます。変数の一般的な構文は以下のいずれかです。

\$c

または

\$(str)

ここで、*c* は 1 つの文字であり、*str* は任意の長さの文字列です。すべてのパラメーターには、1 つの文字とロング・ネームの両方があります。認識されるパラメーターのリストは、イベントによって異なります。

ignore、**echo**、**printEvent**、**sound**、および **shell** アクションは、それぞれ **ignore**、**echo**、**printEvent**、**sound**、および **shell** という名前が付けられているコマンドに想定される機能を実行します。例外事項として、**sound** コマンドは、SGI マシン専用にインプリメントされテストされたものです。

現在認識される変数は、*soundDirectory* と *soundCommand* のみです。

フラグ

項目	説明
-bg	xkbevd にそれ自身を fork するように、さらにバックグラウンドで実行するように指示します。
-cfg <i>File</i>	読み取る構成ファイルを指定します。構成ファイルを指定しない場合、 xkbevd は、順番に <i>~/xkb/xkbevd.cf</i> 、そして \$(LIBDIR)/xkb/xkbevd.cf を探します。
-display <i>Display</i>	使用するディスプレイを指定します。存在しない場合、 xkbevd は \$DISPLAY を使用します。
-help	使用方法メッセージを出力します。
-sc <i>Command</i>	音を再生するために使用されるコマンドを指定します。
-sd <i>Directory</i>	サウンド・ファイルのトップレベル・ディレクトリーを指定します。
-synch	すべての X 要求を強制的に同期します。
-v	デバッグ・メッセージを含め、詳細な情報を表示します。 -v を複数指定すると、複数出力されます。

xkbprint コマンド

目的

XKB キーボード記述をプリントします。

構文

```
xkbprint [ -?| -help ] [ -color ] [ -dflts ] [ -diffs ] [ -eps ] [ -fit ] [ -full ] [ -grid Resolution ] [ -if FontName ] [ -label Type ] [ -lc Locale ] [ -level1 ] [ -level2 ] [ -lg Group ] [ -ll Level ] [ -mono ] [ -n Number ] [ -nkg Number ] [ -npk Number ] [ -o File ] [ -R Directory [ -pict Which ] ] Source [ OutputFile ]
```

説明

xkbprint コマンドは、*Source* によって指定された XKB キーボード記述の、プリント可能なまたはカプセル化された PostScript 記述を生成します。*Source* は、コンパイルされたキーマップの **.xkm** ファイルであり、形状の記述または X ディスプレイの指定が入っています。*OutputFile* が指定されると、**xkbprint** はそれに書き込みます。指定されないと、**xkbprint** は出力ファイルを作成し、要求されたフォーマットに応じてソース・ファイルの拡張子を **.ps** または **.eps** に置き換えます。ソースがローカルでない X ディスプレイの場合、例えば **:0** の場合、**xkbprint** はコロンを **-** (ダッシュ) に置き換えて、ディスプレイの指定に適切な接頭部を追加します。ローカル・ディスプレイの場合、**xkbprint** は **server-n** を使用します。ここで、*n* はディスプレイの番号です。

フラグ

項目	説明
-? -help	使用方法メッセージを出力します。
-color	形状ファイルに指定されているカラーを使ってプリントします。デフォルトでは、 xkbprint は、キーボード・イメージを単色でプリントします。
-dfmts	存在していないコンポーネント (キーなど) のデフォルト名を算出しようとしています。
-diffs	明示的にバウンドされたシンボルだけをプリントします。
-eps	カプセル化された PostScript ファイルを生成します。
-fit	キーボード・イメージをページに適合させます。これはデフォルトです。
-full	キーボードをフルサイズでプリントします。
-grid Resolution	キーボード上に <i>Resolutionmm</i> で格子をプリントします。
-if FontName	指定した出力ファイル、または出力ファイルが指定されていない場合は <i>fontName.pfa</i> にダンプするための、内部 PostScript タイプ 1 フォントを指定します。内部フォントがダンプされない場合、キーボード記述はプリントされません。キーにプリントするラベルを指定します。有効なタイプは以下のとおりです。
-label Type	<ul style="list-style-type: none"> • none • name • code • symbols
-lc Locale	KeySyms を解決するロケールを指定します。
-level1	レベル 1 PostScript を生成します。
-level2	レベル 2 PostScript を生成します。
-lg Group	キーボード・グループにシンボルをプリントします。 <i>Group</i> から始めます。
-ll Level	シフト・レベル <i>Level</i> からシンボルをプリントします。
-mono	キーボードのイメージを単色で生成します。これはデフォルトです。
-n Number	プリント部数 (<i>Number</i>) です。
-nkg Number	<i>Number</i> キーボード・グループのシンボルをプリントします。
-npk Number	各ページにプリントするキーボード・イメージの数 <i>Number</i> を指定します。EPS ファイルの場合、これはプリントするキーボード・イメージの合計を指定します。
-o File	出力を <i>File</i> へ書き込みます。
-R Directory	ルート・ディレクトリーとして、 <i>Directory</i> を使用します。すべてのパス名は、 <i>Directory</i> に関連して解釈されます。
-pict Which	可能な場合、キー・シンボル名の代わりに絵文字の使用を制御します。 <i>Which</i> の有効な値は以下のとおりです。
	<ul style="list-style-type: none"> • all • none • common (デフォルト)
-synch	X 要求を強制的に同期します。
-w Level	警告レベルを設定します。
	<ul style="list-style-type: none"> • 0 はすべての警告を無効にします。 • 10 はすべての警告を有効にします。

関連資料:

192 ページの『xkbcomp コマンド』

xlock コマンド

目的

パスワードが入力されるまで、ローカルの X ディスプレイをロックします。

構文

```
xlock [ -batchcount Number ] [ -bg Color ] [ -delay Users ] [ -display Display ] [ -fg Color ] [
-font FontName ] [ -info TextString ] [ -invalid TextString ] [ -mode ModeName ] [ +mono |
-mono ] [ -username TextString ] [ -nice Level ] [ +nolock | -nolock ] [ -password TextString ] [
+remote | -remote ] [ +allowaccess | -allowaccess ] [ +allowroot | -allowroot ] [
+echokeys | -echokeys ] [ +enablesaver | -enablesaver ] [ -help ] [ -saturation Value ] [
-timeout Seconds ] [ +usefirst | -usefirst ] [ +v | -v ] [ -validate TextString ]
```

説明

xlock コマンドは、ユーザーがキーボードからパスワードを入力するまで X サーバーをロックします。

xlock コマンド実行中は、すべての新規サーバー接続が拒否されます。スクリーンセーバーは使用不可になり、マウス・カーソルはオフになり、画面はブランクになりチェンジ・パターンが表示されます。キーあるいはマウス・ボタンを押すと、**xlock** コマンドを始動したユーザーのパスワードの入力を要求するプロンプトが表示されます。

正しいパスワードが入力されると画面がアンロックされ、X サーバーが復元されます。パスワードを入力するときに、Ctrl-U および Ctrl-H がそれぞれ抹消および消去として使用可能となります。ロックされた画面に戻るには、チェンジ・パターンの小アイコン・バージョンをクリックしてください。

適切に機能させるには、オペレーティング・システムがパスワードおよびアクセス制御ファイルへのアクセスを制限しているため、**xlock** は root 権限で実行されなければなりません。**xlock** に root 許可を与えるには、下記のステップを実行します。

1. root としてログインします。
2. **xlock** プログラム・ファイルのあるディレクトリーに移動します。
3. 下記の 2 つのコマンドを実行します。
 - a. **chownroot xlock**
 - b. **chmodu+s xlock**

フラグ

項目	説明
-batchcount <i>Number</i>	バッチごとに行う処理の数を指定します。 <i>Number</i> は、モードによって参照するものが異なります。
qix	同じ色でレンダラーする線の数を指します。
hop	同じ色でレンダラーするピクセルの数を指します。
イメージ	一度に画面にレンダラーする Sun ロゴの数を指します。
swarm	ミツバチの数を指定します。
life および blank	適用されません。
-bg <i>Color</i>	パスワード画面上のバックグラウンドの色を設定します。

項目	説明
-delay <i>Number</i>	モードが、 hopalong ピクセル、 qix 線、 life 生成、 image ビット、および swarm モーションのバッチの間で遅延操作を行う速度をマイクロ秒で設定します。
-display <i>Display</i>	blank モードでは、キーボードとマウスは各遅延の後にのみ検査されるため、このフラグに小さな値を設定することが重要です。遅延にゼロを設定すると、 blank モードは何もしないため、マウスとキーボードの入力検査が間断なくループし、処理装置に余計な負荷がかかります。 X11 ディスプレイをロックするように設定します。 xlock コマンドは、サーバー上で使用できる画面をすべてロックし、ローカル・サーバー (unix:0, localhost:0, :0 など) しかロックできないように制限します (-remote フラグが設定されていない場合)。
-fg <i>Color</i>	パスワード画面上のテキストの色を設定します。
-font <i>FontName</i>	プロンプト画面上で使用されるフォントを設定します。
-help	使用可能なオプションの簡潔な説明を出力します。
-info <i>TextString</i>	情報メッセージを定義します。デフォルトは「Enter password to unlock; select icon to lock」です。
-invalid <i>TextString</i>	パスワード・メッセージを指定します。デフォルトは「Invalid login」です。
-mode <i>ModeName</i>	下記の 6 つのディスプレイ・モードの中から 1 つを指定します。 blank 何も表示しません。 hop <i>Scientific American</i> の 1986 年 9 月号の実平面フラクタルを表示します。 image 複数の Sun ロゴをランダムに表示します。 life Conway のライフ・ゲームを表示します。 qix 線が回転します。 swarm スズメバチの後をミツバチの群れが追いかけます。 xlock プロセスのシステムの nice レベルを設定します。
-nice <i>NiceLevel</i>	パスワード・プロンプト文字列を指定します。デフォルトは Password: です。
-password <i>TextString</i>	カラー・ランプ (色傾斜) の彩度を設定します。値が 0 のときは中間調、1 のときは明彩色が設定されます。値 0.4 はパステル調の色です。
-saturation <i>Value</i>	パスワード画面がタイムアウトになる時間 (秒) を設定します。
-timeoutSeconds	ユーザー名の前に表示するメッセージを指定します。デフォルトは Name: です。
-username <i>TextString</i>	パスワードの妥当性検査中に表示されるメッセージを指定します。デフォルトは「Validating login...」です。
-validate <i>TextString</i>	アクセス制御リストを使用不可にしますが、ローカル・サーバー上ではパスワードの入力を求めます。 xlock を -KILL コマンドで抹消する場合、アクセス制御リストは失われません。
-/+allowaccess	xlock コマンドをリモートで実行するときにもこのフラグが必要です。理由は、アクセス制御リストへのアクセスが制限されているからです。
-/+allowroot	xlock コマンドを始動したユーザーの他に、 root パスワードでもサーバーをアンロックできるようにします。
-/+echokeys	パスワード・プロンプトの入力時にキーを 1 つタイプするたびに、 xlock コマンドに ? (疑問符) 文字を画面にエコーさせます。デフォルトはエコーなしです。
+/-enablesaver	デフォルトのスクリーンセーバーを使用可能にします。遅延パラメーターの値を大きく設定すると、ディスプレイが焼き付きを起こすことがあります。このフラグは、その予防手段として使用できます。
+/-mono	xlock コマンドはモノクローム (黒白) のピクセルを表示します。デフォルトでは、カラー・ディスプレイにはカラーのピクセルを表示します。
+/-nolock	xlock コマンドはパターン表示のみ行い、ディスプレイをロックしません。キーを押すか、マウスをクリックすると、スクリーンセーバーが終了します。
+/-remote	X11 サーバーをリモート・ロックできるようにします。このフラグは注意して使用しなければなりません。このフラグの主な目的は、 xlock コマンドをローカルで実行できない X11 端末をロックすることです。他の人のワークステーションをロックした場合、そのワークステーションをアンロックするには、ロックした人のパスワードが必要になります。 -remote オプションを使っても、他のシェルへの切り替えは使用可能です。
+/-usefirst	パスワード画面を表示させたキー・ストロークを、パスワードの最初の入力文字として使用できるようにします。デフォルトでは、最初のキー・ストロークは無視されます。

項目
+/-v

説明

マイナス記号を接頭部に使用すると、**xlock** コマンドにより、これから使用するオプションを詳細モードで指示できます。デフォルトの接頭部はプラス記号です。

xlsfonts コマンド

目的

X-Window のフォント・リストを表示します。

構文

```
xlsfonts [ -display Host:Display ] [ -l [ l [ l ] ] ] [ -m ] [ -C ] [ -1 ] [ -w Width ] [ -n Columns ] [ -u ] [ -o ] [ -fn Pattern ]
```

説明

xlsfonts コマンドは、指定した *Pattern* パラメーターと一致するフォントをリストします。任意の文字列 (空を含む) を突き合わせるにはワイルドカード文字 "*" (アスタリスク) を使用し、任意の単一の文字を突き合わせるには "?" (疑問符) を使用します。パターンを指定しなければ、"*" が想定されます。

注: 文字 "*" と "?" は、シェルによって展開されないように、引用符で囲まなければなりません。

フラグを使用すると、サーバー、出力する桁数と幅、フォント・リストのサイズ、出力をソートするかどうか、および **ListFonts** の代わりに **OpenFont** を使用するかどうかを指定することができます。

フラグ

注: **xlsfonts** コマンドの **-l** (L の小文字) フラグを使用すると、サーバーを長時間拘束することがあります。これは、単一スレッドの非占有サーバーに典型的なものであり、プログラム・エラーではありません。

項目

-1

説明

リストが単一の桁を使用することを示します。このフラグは **-n 1** フラグと同じです。

-C

リストに複数桁を使用することを示します。このフラグは **-n 0** フラグと同じです。ホスト名とディスプレイ番号を指定して、接続する X サーバーを識別します。

-display *Host:Display*

-fn *Pattern*

xlsfonts でリストするフォント名の *Pattern* を指定します。

-l [l [l]]

(L の小文字) それぞれ、フォントごとに中程度の長さのリスト、長いリスト、および極端に長いリストが生成されることを示します。

-m

長いリストに、フォントごとの最小および最大限界数も出力されることを示します。

-n *Columns*

出力表示に使用する桁数を指定します。デフォルトでは、**xlsfonts** コマンドは **-w Width** フラグで指定した文字数に納まる桁数でフォント名を表示しようとします。

-o

xlsfonts コマンドに対して、**ListFonts** の代わりに **OpenFont** (および、適切であれば **QueryFont**) を実行するように命令します。 **-o** フラグは、一部のスケール付きフォント・システムの場合と同様に、定義済みフォントを **ListFonts** または **ListFontsWithInfo** でリストできない場合に便利です。

-u

出力をソートしないことを示します。

-w *Width*

何桁表示するかを決めるのに使用する幅を文字数で指定します。デフォルトは 79 です。

環境変数

項目	説明
DISPLAY	使用するデフォルトのホスト番号およびディスプレイ番号を取り出します。

例

1. フォントごとに中程度の長さのリストを指定するには、小文字の **L** を使用して下記のように入力します。

```
xlsfonts -l
```

2. フォントごとに 3 桁のリストを指定するには、下記のように入力します。

```
xlsfonts -n 3
```

3. 名前に文字列 **iso8859** が含まれているフォントをすべて表示するには、下記のように入力します。

```
xlsfonts -ll "*"iso8859"*"
```

4. 名前に含まれる **rom1** の後に文字が 1 文字付いているフォントをすべてリストするには、下記のように入力します。

```
xlsfonts rom1"?"
```

このコマンドにより、下記のようなリストが表示されます。

```
rom10 rom11 rom14 rom16 rom17
```

関連資料:

235 ページの『**xset** コマンド』

xmbind コマンド

目的

仮想キーの割り当てを構成します。

構文

```
xmbind [ -display Host:Display:ScreenID ] [ FileName ]
```

説明

xmbind コマンドは、AIXwindows アプリケーション用の仮想キーの割り当てを構成する X Windows System のクライアントです。このアクションは、**mwm** コマンドがその始動時に行うため、**xmbind** クライアントが必要になるのは、**mwm** コマンドが使用されていないときや、**mwm** コマンドを再始動せずに仮想キーの割り当てを変更したいときのみです。ファイルが指定されていると、その内容は仮想キーの割り当てとして使用されます。ファイルが指定されていない場合、ユーザーのホーム・ディレクトリー中の **.motifbind** ファイルが使用されます。このファイルが見つからなければ、**xmbind** コマンドはデフォルトの仮想キーの割り当てをロードします。

フラグ

項目	説明
-display <i>Host:Display:ScreenID</i>	使用するディスプレイを指定します。 -display オプションのパラメーターは下記のとおりです。
<i>Host</i>	ネットワーク上の有効なシステムのホスト名を指定します。状況によって、ローカル・システムまたはリモート・システムのホスト名を指定することができます。
<i>Display</i>	出力の表示先システムのディスプレイ番号 (通常 0) を指定します。
<i>ScreenID</i>	出力の表示先の画面番号を指定します。単一画面システムでは 0 を指定します。

パラメーター

項目	説明
<i>FileName</i>	仮想マウス・イベントと仮想キー・イベントの割り当てが入っているファイルを指定します。

終了状況

このコマンドは、下記の終了値を戻します。

項目	説明
0	正常に完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。

xmkmf コマンド

目的

Imakefile から **Makefile** を作成します。

構文

```
xmkmf [ -a ] [ TopDir [ CurDir ] ]
```

説明

xmkmf コマンドは、サード・パーティーのソフトウェアとともに出荷された **Imakefile** から **Makefile** を作成します。**Imakefile** ファイルが入っているディレクトリー内で、引数または変数を指定せずに **imake** コマンドを呼び出すと、使用中のシステム (X の構築時に **xmkmf** に構成されたシステム) に適切な引数を使用して実行され、**Makefile** が生成されます。

フラグ

項目	説明
<code>-a</code>	最初に現行ディレクトリー内で Makefile を構築してから、 make Makefiles 、 make includes 、および make depend を自動的に実行します。これにより、MIT X 構築ツリーの外側にあるソフトウェアが構成されます。

変数

MIT X 構築ツリーの内で作業中であれば (X の開発者でない限り、ほとんどあり得ませんが)、*TopDir* と *CurDir* を指定してください。

項目	説明
<i>TopDir</i>	現行ディレクトリーから構築ツリーの最上部への相対パス名として指定します。
<i>CurDir</i>	構築ツリーの最上部から現行ディレクトリーへの相対パス名として指定します。

現行ディレクトリーがサブディレクトリーであれば、*CurDir* 変数が必要です。指定しない場合、**Makefile** でサブディレクトリーを構築できなくなります。代わりに *TopDir* 変数を指定すると、**xmkmf** はシステムに何もインストールされていないものと見なして、インストール済みバージョンを使用する代わりに構築ツリー内でファイルを検索します。

関連情報:

imake コマンド

make コマンド

xmwlm コマンド

目的

システム・パフォーマンスまたは WLM メトリックを記録します。

構文

```
xmwlm [ -d recording_dir ] [ -n recording_name ] [ -t trace_level ] [ -L ]
```

説明

xmwlm エージェントは、ローカル・システム・パフォーマンス・メトリックの限定されたセットに対して記録機能を提供します。 **topas** コマンドによって通常、表示されるこれらのメトリックには、共通 CPU、メモリー、ネットワーク、ディスク、およびパーティションがあります。日次の記録は、`/etc/perf/daily` ディレクトリーに保管されます。 **topasout** コマンドを使用して、ロウ ASCII またはスプレッドシート・フォーマットでこれらの記録を出力します。 **xmwlm** エージェントを使用して、ワークロード・マネージメント (WLM) から記録データを提供することもできます。これは、フラグを何も指定せずに **xmwlm** を実行時に使用されるデフォルト・フォーマットです。日次の記録は `/etc/perf/wlm` ディレクトリーに保管されます。 **wlmon** コマンドを使用して、WLM 関連の記録を処理することができます。 **xmwlm** エージェントは、コマンド・ラインまたはユーザー・スクリプトから開始することができます。また、`/etc/inittab` ファイルの終わり近くに置くこともできます。デフォルトで、すべての記録は 24 時間にわたって行われ 7 日間だけ保存されます。 `/usr/lpp/perfagent/daily.cf` ファイル内の **retain** 値を構成して、デフォルトの記録期間を変更することができます。

フラグ

項目	説明
-d <i>recording_dir</i>	記録ファイル用の出力ディレクトリーを指定します。フラグを無指定で xmwl m を実行時のデフォルトの場所は /etc/perf/wlm 、 -L フラグ指定で xmwl m を実行時は /etc/perf/daily です。
-L	topas に類似のメトリックの集合を指定します。メトリック設定をユーザーが構成することはできません。
-n <i>recording_name</i>	記録ファイルの名前を指定します。デフォルトでは、 xmwl m は、 xmwl m.YYMMDD フォーマットで名付けられた記録ファイルを作成します。例えば、 -n myrecording を指定した場合は、記録ファイルの名前は myrecording.YYMMDD となります。
-t <i>trace_level</i>	トレース・レベルを指定します。 xmwl m は、該当の /etc/perf サブディレクトリーの中のログ・ファイルにさまざまな情報を書き込みます。トレース・レベルは、1 から 9 の範囲内で設定することができます。トレース・レベルが高くなると、より多くのトレース・データが生成されます。このトレース・データは、 xmwl m 記録状況の判別およびデバッグ用に役立ちます。ログ・ファイル名は xmwl m.log1 または xmwl m.log2 です。 xmwl m は、あるファイルが最大許容サイズに到達後にこれら 2 つのファイル間で循環します。

xmwl

m によるセッション・リカバリー

xmwlm エージェントが終了し、再始動した場合は、**xmwl**m は、該当の **/etc/perf** サブディレクトリーまたは **-d** フラグで指定されたディレクトリーの中の記録ファイルを調べます。記録ファイルが現在日付で存在する場合は、**xmwl**m がデータをこのファイルに追加し、記録ファイルへの書き込みを続けます。それ以外の場合は、新規記録ファイルが作成されます。

位置

/usr/bin/xmwlm

ファイル

項目	説明
/usr/bin/xmwl m	xmwl m エージェントが入っています。このエージェントは、 perfagent.tools ファイルセットの一部です。

関連情報:

topas コマンド

topasout コマンド

xmodem コマンド

目的

xmodem プロトコルを使用して、ファイルを転送し、非同期伝送中のデータ伝送エラーを検出します。

構文

xmodem { **-s** | **-r** } *FileName*

説明

xmodem シェル・コマンドは、*FileName* パラメーターで指示されたファイルを、**xmodem** プロトコルを使って転送するために、非同期端末エミュレーション (ATE) プログラムと一緒に使用します。

xmodem プロトコルは 8 ビットの転送プロトコルであり、データ伝送エラーを検知した後でデータを再送することができます。データを受け取る準備ができたことを知らせるシグナルをリモート・システムが送信するまで、データ送信側のワークステーションは待機します。

受信側システムはデータを受け取ると、送信側システムに肯定応答を戻します。ATE プログラムでは、受信側システム時刻が、ファイル転送が開始されてから 90 秒以内にデータを受け取らなければ、タイムアウトになります。

xmodem コマンドを使って行う送受信は相補的操作です。一方のシステムが受信する設定になっているとき、もう一方のシステムは必ず送信する設定になっていなければなりません。リモート・システムの **xmodem** コマンドは、ローカル・システムの「ATE Connected Main Menu (ATE 接続メインメニュー)」からの **send** サブコマンドあるいは **receive** サブコマンドと組み合わせて使用します。

xmodem ファイル転送に割り込むには、Ctrl-X キー・シーケンスを押します。

注:

1. DOS オペレーティング・システムでは、ASCII ファイルの各行の終了には、改行文字と復帰 (Ctrl-M) 文字が使用されます。UNIX では、ASCII ファイルの各行の終了には、改行文字のみが使用されます。DOS ファイルが AIX に転送される時も復帰文字は保持されます。vi テキスト・エディターを使用して次のサブコマンドで、関係のない Ctrl-M 文字を除去することができます。

```
:%s/<Ctrl-V><Ctrl-M>//
```

ここで <Ctrl-V> と <Ctrl-M> は、それぞれ入力される単一の制御文字を表します。しかし、Ctrl-V は ATE のデフォルト MAINMENU_KEY であるため、ATE でログインしている間は、vi サブコマンドを発行するために ATE のデフォルトを変更する必要があります。

2. **xmodem** ファイル転送プロセスでは、最後に転送されるパケットに 128 バイト長にするための Ctrl-Z 文字が追加されます。そのため、転送されるほとんどのファイルの終わりに、Ctrl-Z 文字が追加されることとなります。DOS オペレーティング・システムでは、ASCII ファイルの終了に Ctrl-Z 文字が使用されます。したがって、DOS から AIX に転送されるあらゆるファイルは、少なくとも 1 つの Ctrl-Z 文字で終了します。この余分の Ctrl-Z 文字を、vi テキスト・エディターを使用して除去することができます。

フラグ

項目	説明
-r	ローカル・ワークステーションからデータを受信します。
-s	ローカル・ワークステーションヘデータを送信します。

例

xmodem プロトコルを使ってファイルを送る場合

xmodem プロトコルを使ってファイル *myfile* を送信するには、**ate** コマンドと、**connect** または **directory** サブコマンドを使ってリモート・システムへ接続します。

1. リモート・システムへログインしてから、ローカル・システム上の ATE へ戻るために MAINMENU_KEY (通常は Ctrl-V キー・シーケンス) を押す前に、シェル・コマンド・ラインで下記のように入力します。

```
xmodem -r myfile
```

xmodem プロトコルが、リモート・システム上で受信モードを開始します。

2. MAINMENU_KEY を押してローカル・システム上の ATE へ戻ります。

「ATE Connected Main Menu」が表示されます。

3. 「ATE Connected Main Menu」上のプロンプトで、下記のように **send** サブコマンドを入力します。

```
s myfile
```

send サブコマンドはローカル・システムに対して、リモート・システムへ myfile を送るように指示します。ファイル転送後、「ATE Connected Main Menu」が表示されます。

xmodem プロトコルを使ってファイルを受け取る場合

xmodem プロトコルを使ってリモート・システムからファイル infile を受け取るには、**ate** コマンドと、**connect** あるいは **directory** サブコマンドを使ってリモート・システムへ接続します。

1. リモート・システムへログインしてから、ローカル・システム上の ATE へ戻るために MAINMENU_KEY (通常は Ctrl-V キー・シーケンス) を押す前に、シェル・コマンド・ラインで下記のように入力します。

```
xmodem -s infile
```

xmodem プロトコルが、リモート・システム上で送信モードで開始されます。

2. MAINMENU_KEY を押してローカル・システム上の ATE へ戻ります。

「ATE Connected Main Menu」が表示されます。

3. 「ATE Connected Main Menu」上のプロンプトで、下記のように **receive** サブコマンドを入力します。

```
r infile
```

receive サブコマンドはローカル・システムに対して、リモート・システムから infile を受け取るように指示します。ファイル転送後、「ATE Connected Main Menu」が表示されます。

ファイル

項目	説明
ate.def	ATE のデフォルト値が入っています。

関連情報:

ate コマンド

ATE のデフォルト・ファイルの編集

非同期通信端末エミュレーション

xmodmap コマンド

目的

X サーバー内のキー・マップを変更します。

構文

```
xmodmap [ -display Display ] [ -e Expression ] [ -grammar | -help ] [ -n ] [ -pk ] [ -pke ]  
[ -pm ] [ -pp ] [ -quiet | -verbose ] [ FileName ]
```

説明

xmodmap コマンドは、クライアント・アプリケーションがイベント・キー・コードをキー記号へ変換するために使用するキーボード修飾子マップおよびキー・マップ・テーブルを編集および表示します。このコマンドは通常、セッション始動スクリプトから実行し、キーボードの構成をユーザーの好みに合わせます。

キー・コード式が評価されるたびに、サーバーはすべてのクライアントに **MappingNotify** イベントを生成します。変更はまとめて一回でバッチ処理される必要があります。キーボード入力を受け取っても **MappingNotify** イベントを無視するとクライアントは、キーボード・マッピングの変更にまったく気付きません。

FileName パラメーターは、実行される **xmodmap** コマンド式を含むファイルを指定します。このファイルは通常、ユーザーのホーム・ディレクトリーに **.xmodmaprc** という名前で保管されています。ファイルが指定されなければ、**stdin** を入力とします。

xmodmap コマンド・プログラムは式のリストを読み取り、実行を試みる前にすべての式を解析します。これによって、名前が矛盾していないかどうかを心配することなく、本来の形で再定義されるキー・シンボルを参照することができます。

項目

add

説明

行が読み取られるたびにキー・シンボル名が評価されます。これによってキーが再割り当てされたかどうかを心配することなく修飾子からキーを除去することができます。

add *ModifierName* = *KeySymbolName*...

与えられたキー・シンボルを指定された修飾子マップに追加します。キー・シンボル名は、キーをスワップする式を書きやすくするために、すべての入力式が読み取られた後に評価されます。

clear *ModifierName*

与えられた修飾子の修飾子マップの中のすべてのエントリーを除去します。

keycode *Number* = *KeySymbolName*...

このとき有効な名前は **Shift**、**Lock**、**Control**、**Mod1**、**Mod2**、**Mod3**、**Mod4**、**Mod5** です (他のすべての名前と異なり、修飾子名では英大文字と小文字の区別はありません)。例えば、**clear Lock** は **shift lock** 修飾子にバインドされたすべてのキーを除去します。

キー・シンボルのリストを指定されたキー・コードに割り当てます。このキー・コードは、10 進数、16 進数、または 8 進数で指定したり、**/usr/lpp/X11/Xamples/demos** ディレクトリーにある **xev** プログラムを実行して判別することができます。通常は与えられたコードに対してキー記号が 1 つだけ割り当てられます。

keySYM *KeySymbolName* = *KeySymbolName*...

左辺の *KeySymbolName* はマッチング・キー・コードに変換され、対応するキー・コード式のセットを実行するために使用されます。 **keySYM** 名のリストは、**keySYM** データベース **/usr/lib/X11/XkeySYMDB** あるいはヘッダー・ファイル **X11/keySYMdef.h** (XK_ 接頭部なし) にあります。複数キーに同一の **keySYM** をバインドしているときは、この式はマッチング・キー・コードに対して実行されることに注意してください。

項目	説明
pointer = default	ポインター・マップをデフォルト設定へ戻します (ボタン 1 によって 1 のコードが、ボタン 2 によって 2 のコードが設定され、... というように設定されます)。
pointer = Button1 Button2 Button3...	ポインター・マップを設定して、指定されたボタン・コードを入れます。リストは必ず最初の物理ボタンから始められます。
remove ModifierName = KeySymbolName...	与えられた keysym を含むキーすべてを指定された修飾子マップから除去します。 add とは異なり、 keysym 名は、その行が読み取られるものとして評価されます。これは、再配置されたかどうかにかかわらず修飾子からのキーを除去できるようにします。

(感嘆符) で始まる行はコメントとして処理されます。

修飾子キーのバインドを変更したい場合も、そのキーを適切な修飾子マップから除去する必要があります。

フラグ

項目	説明
-display Display	使用するホストとディスプレイを指定します。
-e Expression	実行する式を指定します。コマンド・ラインからいくつでも式を指定することができます。
-grammar	-e Expressions フラグを付けてファイルの中で使用する、式の文法について説明するヘルプ・メッセージを、標準エラー上に出力します。
-help	コマンド・ライン引数の簡単な説明を、標準エラー上に出力します。 xmodmap コマンドに、処理されない引数が与えられた場合、この動作を必ず実行します。
-n	xmodmap コマンドによってマッピングを変更しないこと、ただしこのフラグを指定したときには何が行われるかを表示することを示します。
-pk	現在のキー・マップ・テーブルを標準出力へ出力することを示します。
-pke	現在のキー・マップ・テーブルを、 xmodmap で戻される式のフォーマットで標準出力へ出力することを示します。このフラグは X11R5 に固有のものでした。
-pm	現在の修飾子マップを標準出力へ出力することを示します。
-pp	現在のポインター・マップを標準出力へ出力することを示します。
-quiet	詳細ログをオフにします。これはデフォルトです。
-verbose	入力を解析しながら、 xmodmap コマンドがログ情報を出力することを示します。

例

1. 下記のコマンドでは、ポインター (3 つボタン) 上で左手の人差し指を使って第 1 ボタンが押せるように生成したボタン・コードを、反転させることができます。

```
xmodmap -e "pointer = 1 2 3 4 5"
```

2. 下記のコマンドは、メタを複数言語キー (Compose Character と呼ばれることもあります) へ付加します。またこれは、メタ・キーを必要とするアプリケーションにはキー・コードだけが必要であって、キー・マップ・テーブルの最初の欄にあるキー記号は必要としない、という事実に基づいています。これは、**Multi_key** (デフォルト修飾子マップを含む) を検索するアプリケーションはいかなる変更も通知しないという意味になります。

```
keysym Multi_key = Multi_key Meta_L
```

3. コンマおよびピリオド・キーがシフトされたとき不等号文字を自動生成するには、下記のスクリプトでコンマとピリオドの割り当てを再設定します。

```
!
! make shift-, be < and shift-. be >
!
keysym comma = comma less
keysym period = period greater
```

4. 制御キーとシフト・ロック・キーの位置を入れ替えるには、下記のスクリプトを使います。

```

!
! Swap Caps_Lock and Control_L
!
remove Lock = Caps_Lock
remove Control = Control_L
keysym Control_L = Caps_Lock
keysym Caps_Lock = Control_L
add Lock = Caps_Lock
add Control = Control_L

```

xmpeek コマンド

目的

xmpeek コマンドを使用すると、任意のホストに照会して、その **xmtopas** デーモンの状況について問い合わせることができます。

構文

```
xmpeek [ -a | -l ] [ hostname ]
```

説明

xmpeek コマンドは、現在 **xmtopas** デーモンで定義済みの計測データ (統計セット) を持つデータ・コンシューマーを一覧表示したり、**xmtopas** デーモンが認識しているデータ・コンシューマーをすべて一覧表示したりするために使用されます。また、この **xmpeek** コマンドは、ある特定のホストについてすべての使用可能な SPMI 統計情報を印刷するためにも使用されます。

フラグ

項目	説明
-a	このフラグが指定されると、デーモンが認識しているデータ・コンシューマーごとに 1 行ずつリストされます。このフラグが使用されていない場合は、現在このデーモンで定義済みの計測データ (統計セット) を持つデータ・コンシューマーのみがリストされます。このフラグはオプションです。
-l	リモート・ホストの入手可能な統計をすべてリストします。このフラグはオプションです。
<i>hostname</i>	ホスト名が指定されると、指定されたホスト上のデーモンが要求されます。ホスト名が指定されない場合は、ローカル・ホスト上のデーモンが要求されます。

例

xmpeek プログラムからの出力の例を示します。

```

Statistics for xmtopas daemon on *** birte ***
Instruments currently defined: 1
Instruments currently active: 1
Remote monitors currently known: 2
--Instruments--- Values Packets
                        Internet Protocol
Defined Active Active Sent Address Port Hostname
-----
1 1 16 3,344 129.49.115.208 3885 xtra

```

xmpeek からの出力には、以下の 2 つの形式があります。

1 番目の形式は、**xmtopas** デーモンがデータ・コンシューマー・プログラムを何もフィードしていないことを知らせる 1 行です。この形式は、デーモンで統計セットが定義されておらず、コマンド・フラグが指定されていない場合に使用されます。

2 番目の形式には、少なくとも前の例で示される情報と同様な情報は含まれます。ただし、**-a** フラグが使用されている場合、またはデーモンで定義済みの計測データ (統計セット) をデータ・コンシューマーが少なくとも 1 つ持っている場合のみ、ホスト **xtra** 上のデータ・コンシューマーに対して 1 明細行が表示されます。**xmpeek** は Remote Statistics Interface (RSI) API を使用してデーモンに接続するため、それ自身がデータ・コンシューマーとして表示されることに注意してください。そのため、出力には常に少なくとも 1 つの既知のモニターが表示されます。

固定出力には、デーモンが稼働中のホストの名前が最初に表示されます。次の 3 行には、デーモンの現在の状況がすべて表示されます。上の例では、計測データが 1 つだけ定義されており、現在アクティブであることが分かります。また、2 つのデータ・コンシューマーがデーモンに認識されていますが、そのうちの 1 つだけが **birte** 内のデーモンで定義済みの計測データを持っていることも分かります。明らかに、この出力は **-a** フラグなしで作成されたものです。

活動状況がより多い例として、**xmpeek** からの出力例を以下に示します。この出力は、次のコマンドを使用して作成されたものです。

```
xmpeek -a birte
```

一部の明細行には、定義済みの計測データがゼロと表示されていることに注意してください。そのような明細行は、データ・コンシューマーから **are_you_there** メッセージを受け取ったが、状態が定義されたことがないか、または以前の定義済み状態が消去されたことを示します。

```
Statistics for smeared daemon on *** birte ***
Instruments currently defined: 16
Instruments currently active: 14
Remote monitors currently known: 6
--Instruments--- Values Packets Internet Protocol
Defined Active Active Sent Address Port Hostname
8 8 35 10,232 129.49.115.203 4184 birte
6 4 28 8,322 129.49.246.14 3211 umbra
0 0 0 0 129.49.115.208 3861 xtra
1 1 16 3,332 129.49.246.14 3219 umbra
0 0 0 0 129.49.115.203 4209 birte
1 1 16 422 129.49.115.208 3874 xtra
-----
16 14 95 22,308
```

同じホスト名が 2 回以上表示される場合があることに注意してください。これは、**xmperf** のすべての実行中のコピーとその他のすべてのアクティブなデータ・コンシューマー・プログラムがカウントされ、別個のデータ・コンシューマーとして取り扱われるためです。それぞれは、**xmpeek** の出力に示されているように、UDP パケット用に使用されるポート番号によって識別されます。

2 番目の明細行は、ホスト **umbra** 上の 1 つの特定のモニターに 6 つの計測データが定義されているが、4 つのみがアクティブであることを示しています。これは、リモート **xmperf** コンソールがオープンされていたが、現在はクローズされている場合に起こる可能性があります。**xmperf** コンソールをクローズすると、**xmperf** メインウィンドウの「モニター」メニューのままになり、コンソールの計測データの定義はデータ・サプライヤー・デーモンのテーブル内にそのまま残りますが、計測データは非アクティブになります。

項目	説明
-i <i>min_remote_interval</i>	データ送信が可能な最小の間隔を、ミリ秒単位で定義します。デフォルトは 500 ミリ秒です。100 ミリ秒から 5,000 ミリ秒までの値を指定できます。指定された値は、100 ミリ秒の倍数に丸められます。最小リモート間隔が指定されると、データ送信のためのすべての要求は、この値の倍数に丸められるようになります。
-l <i>remove_consumer_timeout</i>	『 xmtopas の開始および停止 (Life and Death of xmtopas)』のセクションに記述されているように、統計データの送信が停止した後の time_to_live を設定します。続いて、分数を指定する必要があります。値が 0 (ゼロ) 分の場合、デーモンは永久に活動状態のままになります。デフォルトの time_to_live は 15 分です。
-m <i>supplier_timeout</i>	ダイナミック・データ・サプライヤーが活動状態のときに、この値により非活動状態の秒数が設定されます。DDS の非活動状態がここで設定済みの秒数経過すると、SPMI は DDS が停止していると見なします。このタイムアウト値を超えたときには、 SiShGoAway フラグが共有メモリー領域内に設定され、SPMI はその領域から切断されます。このフラグが設定されない場合、タイムアウト期間は 90 秒に設定されます。
-p <i>trace_level</i>	タイムアウト期間のサイズは、共通の SPMI 共有メモリー領域内に保持されます。保管される値は、 xmtopas コマンドを含む、任意のデータ・コンシューマー・プログラムが要求する最大値です。 トレース・レベルを設定します。これにより、 /etc/perf/xmtopas.log1 ログ・ファイルまたは /etc/perf/xmtopas.log2 ログ・ファイルに書き込まれるイベントのタイプが決まります。このフラグの後に、0 から 9 までの数字を付ける必要があります。9 が最も詳細なトレース・レベルです。デフォルトのトレース・レベルは 0 (ゼロ) です。この場合、イベントのトレースおよびログギングは使用不可ですが、エラー・メッセージはログに記録されます。
-s <i>max_logfile_size</i>	ログ・ファイルの概算の最大サイズを指定します。少なくとも time_to_live 分毎に、現在アクティブなログ・ファイルが <i>max_logfile_size</i> の値よりも大きいかどうか検査されます。この値よりも大きい場合、現在のログ・ファイルはクローズされ、代替ログ・ファイル (まず長さがゼロにリセットされます) に対してログギングが続行されます。ログ・ファイルには、 /etc/perf/xmtopas.log1 と /etc/perf/xmtopas.log2 の 2 つがあります。デフォルトの最大ファイル・サイズは 100,000 バイトです。 <i>max_logfile_size</i> の値を 5,000 バイトより小さくしたり、10,000,000 バイトより大きくしたりすることはできません。
-t <i>keep_alive_limit</i>	<i>keep_alive_limit</i> 値の後に、60 から 900 までの秒数 (1 分から 15 分) を設定する必要があります。デフォルトは 300 秒 (5 分) です。
-x <i>xmtopas_execution_priority</i>	xmtopas コマンドの実行優先順位を設定します。お客様の環境で、 xmtopas コマンドのデフォルトの実行優先順位が不適切な場合は、このオプションを使用してください。一般に、このデーモンには可能な限り高い実行優先順位を設定する必要があります (より小さな数値を設定すれば、実行優先順位はより高くなります)。

IBM RS/6000® システム以外のシステムでは、**-x** フラグを使用して、**xmtopas** コマンドの適切な優先順位を設定します。適切な優先順位の値は -20 から 19 です。デフォルトは -20 です。

xmtopas コマンドが使用するファイル

xmtopas.res ファイルには、以下のエントリーを指定できます。

```
docec:<arguments>
docluster:cluster=<cluster configuration file>
```

例

```
docec: availmem=5 unavailmem=2
docluster: cluster=/etc/perf/xmtopasagg.cf
```

以下の新しいフィールドが **xmtopas.res** ファイルの **docec** エントリーに追加され、ハードウェア管理コンソール (HMC) の詳細を入手できるようになりました。

```
managedsys=[Managed system name under which this partition is configured]
hmc=[HMC name under which this partition is configured]
```

HMC プラットフォームで自動照会を構成できない場合は、ローカル・パーティションに使用できないグローバル・データ・フィールドを、以下のオプションを使用して設定することができます。

```
availmem = [Total amount of memory allocated to all partitions, in GB]
unavailmem = [Total amount of memory unallocated from the HMC, in GB]
availprocessor = [Total number of physical processors allocated for all partitions]
unavailprocessor = [Total number of physical processors unallocated from the HMC]
poolsize = [Defined Pool Size required if HMC Processor Utilization Authority restricts access]
partitions = [Number of partitions defined on the HMC]
reconfig = [Number of seconds between checking for HMC configuration changes.
Allowed values are 30, 60, 90, 120, 180, 240, 300 seconds.
The default is 60 seconds.]
```

例

```
docec: hmc=hmcmac.in.ibm.com managedsys=cec1
```

xntpd デーモン

目的

Network Time Protocol (NTP) デーモンを始動します。

構文

```
xntpd [ -a ] [ -b ] [ -d ] [ -D Level ] [ -m ] [-x] [ -c ConfigFile ] [ -e AuthenticationDelay ] [
-f DriftFile ] [ -k KeyFile ] [ -l LogFile ] [ -o TraceFile ] [ -p pidFile ] [ -r BroadcastDelay ] [
-s StatsDirectory ] [ -t TrustedKey ] [ -v SystemVariable ] [ -V SystemVariable ]
```

説明

xntpd デーモンは、インターネット標準タイム・サーバーに準拠した UNIX システム時刻を設定し、保守します。**xntpd** デーモンは、RFC 1305 で定義されているように Network Time Protocol (NTP) バージョン 3 標準の完全インプリメンテーションであり、また、RFC 1059 および RFC 1119 で定義されているようにバージョン 1 および 2 のサーバーとの互換性も保持します。**xntpd** デーモンは、すべての計算を固定小数点数演算で行い、浮動小数点コードを必要としません。

xntpd デーモンは、始動時に構成ファイルから読み取りをします (デフォルト値は `/etc/ntp.conf`)。コマンド・ラインから構成ファイル名を上書きすることができます。また、(限定された) 作業構成をすべてコマンド・ラインで指定して、構成ファイルの必要をなくすることもできます。**xntpd** デーモンを、実行時にブロードキャストを `listen` して、すべてのピアを判別するブロードキャストまたはマルチキャスト・クライアントとして構成する場合、このメソッドを使用します。**xntpd** デーモンの内部変数を `ntpq` コマンド (Network Time Protocol (NTP) 照会プログラム) によって表示することができます。**xntpd** コマンドによって構成オプションを変更することができます。

xntpd デーモンは、対称的な、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いくつかのモードで操作されます。ブロードキャスト/マルチキャスト・クライアント

トは、リモート・サーバーを自動的に発見し、片方向の遅延訂正ファクターを計算し、それ自体を自動的に構成することができます。このモードによって構成ファイルまたはその環境に特定の構成の詳細を指定せずにワークステーションのグループを展開することができます。

注: クライアント・モードで操作する場合、ローカル・システム時刻の 1000 秒以内に構成サーバーがない場合、**xntpd** デーモンはエラーで終了します。**xntpd** を始動する前に正しくないスキュー・システムの時間を設定するには、**date** コマンドまたは **ntpdate** コマンドを使用します。

フラグ

項目	説明
-a	認証モードで実行されます。
-b	ブロードキャスト NTP を listen し、使用可能な場合、それらに同期します。
-c ConfigFile	代替構成ファイルの名前を指定します。
-d	デバッグ・モードを指定します。フラグは、複数回 (最大 10 回まで) 発生する可能性があり、発生たびに詳しく表示されます。
-D Level	デバッグ・レベル (1 から 10 までの値) を直接指定します。
-e AuthenticationDelay	このコンピューターで NTP 暗号化フィールドを計算するのに必要な時間を秒単位で指定します。
-f DriftFile	ドリフト・ファイルの位置を指定します。
-k KeyFile	NTP 認証キーを含むファイルの位置を指定します。
-l LogFile	(小文字 L) syslog ヘログ記録する代わりにログ・ファイルの使用法を指定します。
-m	マルチキャスト・メッセージを listen し、使用可能な場合、それらに同期します。マルチキャスト・アドレス 224.0.1.1 を想定します。
-o TraceFile	トレース・ファイル名 (デフォルトは stderr) を指定します。
-p pidFile	デーモンの処理 id を記録するファイルの名前を指定します。デフォルト値はありません。
-r BroadcastDelay	調整手順が失敗した場合、デフォルトの遅延 (秒数) を指定します。通常、 xntpd デーモンは、ブロードキャスト/マルチキャスト・サーバーとクライアントの間のネットワークの遅延を自動的に補正します。
-s StatsDirectory	統計情報ファイルの作成に使用するディレクトリーを指定します。
-t TrustedKey	トラステッド鍵リストに指定のキー番号を追加します。
-v SystemVariable	指定のシステム変数を追加します。
-V SystemVariable	デフォルトでリストされた指定のシステム変数を追加します。
-x	時間の微調整を行います (随時調整)。

リファレンス時計サポート

構成のために **xntpd** デーモンは、可能な限り通常の NTP ピアに似た方法でリファレンス時計を扱います。通常のピアと区別するために無効な IP アドレスを使用しますが、通常のピアと同じように、アドレスによってリファレンス時計を参照します。AIX は、システム時計 (1 型) に基づいて 1 つの型のリファレンス時計をサポートします。

リファレンス時計アドレスのフォーマットは、**127.127.Type.Unit** です。ここで、**Type** は、時計型を示す整数で、**Unit** は、型特定の装置番号を示します。**HostAddress** が時計アドレスである構成ファイルでサーバー・ステートメントを使用してリファレンス時計を構成します。キー・オプション、バージョン・オプション、および **ttl** オプションは、リファレンス時計サポートには使用されません。

リファレンス時計サポートは、リファレンス時計を特殊な方法で構成する **fudge** コマンドを提供します。このコマンドのフォーマットは、下記のとおりです。

```
fudge 127.127.Type.Unit [ time1 Seconds ] [ time2 Seconds ] [ stratum Integer ] [ refid Integer ] [ flag1 0 | 1 ] [ flag2 0 | 1 ] [ flag3 0 | 1 ] [ flag4 0 | 1 ]
```

time1 オプションおよび **time2** オプションは、固定小数点秒数で示されており、いくつかの時計ドライバーで調整定数として使用されます。

stratum オプションは、0 から 15 までの範囲の数で、標準外操作層を時計に割り当てる場合に使用します。**xntpd** デーモンは、各ピアの層に 1 を追加するために、通常、1 次サーバーが第 1 層を表示します。技術的バックアップを提供するには、**stratum** オプションを使用して、リファレンス時計層をゼロより大きく指定します。注釈がある場合を除き、このオプションはすべての時計ドライバーに適用されます。

refid オプションは、1 から 4 文字までの範囲の ASCII 文字列であり、標準外リファレンス ID を時刻に割り当てる場合に使用します。

バイナリー・フラグ: **flag1**、**flag2**、**flag3**、および **flag4** は、時計ドライバーのカスタマイズ用フラグです。これらの値の変換処理およびこれらの値が使用されるかどうかを決めるのが特定の時計ドライバーに必要なファンクションです。

終了状況

このコマンドは、下記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 権限がなければなりません。

監査イベント: N/A

例

1. **xntpd** デーモンを始動するには、下記を入力します。
`startsrc -s xntpd`
2. **xntpd** デーモンを停止するには、下記を入力します。
`stopsrc -s xntpd`
3. **xntpd** デーモンの実行中に認証キー付きファイル `/etc/ntp.new.keys` を使用するには、下記を入力します。
`/usr/sbin/xntpd -k /etc/ntp.new.keys`

ファイル

項目	説明
<code>/usr/sbin/xntpd</code>	xntpd デーモンを含みます。
<code>/etc/ntp.conf</code>	デフォルトの構成ファイルを含みます。
<code>/etc/ntp.drift</code>	デフォルトのドリフト・ファイルを含みます。
<code>/etc/ntp.keys</code>	デフォルトのキー付きファイルを含みます。

関連資料:

216 ページの『`xntpd` コマンド』

関連情報:

`ntp` コマンド

xntpd コマンド

目的

Network Time Protocol デーモン **xntpd** 用の照会/制御プログラムを始動します。

構文

```
xntpd [ -i ] [ -l ] [ -n ] [ -p ] [ -s ] [ -c SubCommand ] [ Host ... ]
```

説明

xntpd コマンドは、現在の状態について **xntpd** デーモンに照会し、その状態を変更するよう要求します。このコマンドは、対話式モードまたはコマンド・ライン引数の使用によって実行されます。**xntpd** コマンド・インターフェースは、広範囲な状態および統計情報を表示します。始動時に **xntpd** デーモンの構成ファイルを使用して指定できる、ほとんどすべての構成オプションは、実行時に **xntpd** コマンドを使用しても指定できます。

1 つ以上の要求フラグを指定して **xntpd** コマンドを入力した場合、指定の各ホスト（またはデフォルトでローカル・ホスト）で実行中の NTP サーバーは、各要求を受け取ります。要求フラグをまったく入力しない場合、**xntpd** コマンドは標準入力からのコマンドを読み取り、それを最初に指定されたホストまたはデフォルトのローカル・ホストで実行中の NTP サーバーで実行しようとしています。標準入力端末である場合、サブコマンドを要求するプロンプトを出します。

xntpd コマンドは、NTP モード 7 パケットを使用して、NTP サーバーと通信し、それを許可するネットワーク上で任意の互換性のあるサーバーを照会することができます。

xntpd コマンドは、要求を再送しようとはせず、リモート・ホストが適切な時間内に応答しなければ、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、照会が指定のホストに即時に送信されます。それ以外の場合、**xntpd** コマンドは、標準入力から対話式フォーマット・コマンドを読み取ろうとします。

フラグ

項目	説明
-c SubCommand	対話式フォーマット・コマンドを指定します。このフラグは、 <i>SubCommand</i> をコマンド・リストに追加して、指定のホストを実行します。複数の -c フラグを入力することができます。
-i	対話式モードを指定します。標準出力はプロンプトを表示し、標準入力からコマンドを読み取ります。
-l	(小文字 L) サーバーが認識できるピアの一覧を表示します。これは、 listpeers サブコマンドと同じです。
-n	すべてのホスト・アドレスを正規ホスト名ではなく、ドット表記 10 進数フォーマット (0.0.0.0) で表示します。
-p	サーバーが認識できるピアの一覧およびその状態の要約を表示します。これは、 peers サブコマンドと同じです。
-s	サーバーが認識できるピアの一覧およびその状態の要約を -p フラグとは異なるフォーマットで表示します。これは dmpeers サブコマンドと同じです。

パラメーター

項目	説明
<i>Host ...</i>	ホストを指定します。

xntpd 内部サブコマンド

いくつかの対話式フォーマット・サブコマンドは、NTP モード 7 の要求をサーバーに送信しない **xntpd** コマンド内ですべて実行することができます。下記のサブコマンドは、**xntpd** 照会プログラムの実行中のみ使用することができます。

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、キーワードおよびそれに続く 0 値から 4 値までの引数から成ります。サブコマンドを一意的に識別するのに十分な文字を全キーワードから入力するだけでかまいません。サブコマンドの出力は標準出力に行先指定されますが、より大符号 (>) およびそれに続くファイル名をコマンド・ラインに付加して、個別サブコマンドの出力をファイルにリダイレクトすることができます。

項目	説明
? [<i>SubCommand</i>]	コマンドの使用に関する情報を表示します。 <i>SubCommand</i> なしで使用した場合、 xntpd コマンド・キーワードの一覧をすべて表示します。 <i>SubCommand</i> とともに使用した場合、コマンドのファンクションおよび使用に関する情報を表示します。
help [<i>SubCommand</i>]	? [<i>Subcommand</i>] サブコマンドと同じです。
delay <i>Milliseconds</i>	認証が必要な要求に含まれるタイム・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い遅延ネットワーク・パスまたは同期されていない時計を持つマシン間の信頼性のないサーバー再構成を使用可能にします。このサブコマンドを引数なしに入力した場合、このサブコマンドの現在の設定が表示されます。
host <i>HostName</i>	照会の送信先のホストを指定します。 <i>HostName</i> は、ホスト名または数値アドレスのいずれかの可能性があります。このサブコマンドを引数なしに入力した場合、このサブコマンドの現在の設定が表示されます。
hostnames yes no	ホスト名 (yes) または数値アドレス (no) を表示するかどうかを指定します。デフォルト値は、 -n フラグが使用されない限り yes です。このサブコマンドを引数なしに入力した場合、このサブコマンドの現在の設定が表示されます。
keyid <i>Number</i>	構成要求の認証に使用するサーバー・キーの数を指定します。このサブコマンドを引数なしに入力した場合、このサブコマンドの現在の設定が表示されます。
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出します。
quit	xntpd 照会プログラムを終了します。
timeout <i>Milliseconds</i>	サーバー照会への応答のためのタイムアウト期間を指定します。デフォルト値は、8000 ミリ秒です。このサブコマンドを引数なしに入力した場合、このサブコマンドの現在の設定が表示されます。

照会サブコマンド

xntpd 照会サブコマンドは、要求を含む NTP モード 7 パケットを結果的にサーバーに送信します。これらのサブコマンドは、読み取り専用です (サーバー構成状態を変更しません)。

項目		説明
clkbug <i>ClockPeerAddress</i> [<i>Addr2</i>] [<i>Addr3</i>] [<i>Addr4</i>]		リファレンス時計ドライバーのデバッグ情報を表示します。いくつかの時計ドライバーは、ドライバー・ソースのコピーのコントロールなしにはほとんどデコードできないこの情報を提供します。
clockbug <i>ClockPeerAddress</i> [<i>Addr2</i>] [<i>Addr3</i>] [<i>Addr4</i>]		ピア時計に関する情報を表示します。取得した値は、限度を超えたファクターの設定およびその他の時計パフォーマンス情報に関する情報を提供します。
dmpeers		サーバーが状態を保守するピアの一覧を、その状態の要約とともに表示します。左端の列の文字を除いて、 peers サブコマンドの出力と同一です。文字は、時計選択アルゴリズムの最終ステージで取り込まれたピアの側にのみ表示されます。
		左端の列の文字には、下記のものがあります。
		<ul style="list-style-type: none"> . このピアが falseticker 検出でキャストオフされたことを示します。 + ピアが成功したことを示します。 * サーバーが現在同期中のピアを示します。
iostats		入出力モジュールで保守される統計情報カウンターを表示します。
kerninfo		カーネル・フェーズ・ロック・ループ操作パラメーターを表示します。この情報は、生成中のホストのカーネルが精度タイム・キーピング・ファンクション用に特別に変更された場合のみ使用可能です。
listpeers		サーバーが状態を保守しているピアの簡単な一覧を表示します。ここには、すべての構成されたピアの関連付けおよび今後、同期化される可能性があるサーバーが解釈する層を持つピアが含まれます。
loopinfo [<i>online</i> <i>multiline</i>]		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・システム時計を調整する 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、ローカル時計の百万分率 (ppm) 頻度エラーです。 poll adjust は、フェーズ・ロック・ループの強固さ (変更への抵抗) およびそれが発振器ドリフトに適応できるスピードをコントロールします。 watchdog timer 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。 online オプションおよび multiline オプションは、この情報を表示するフォーマットを指定します。 multiline オプションは、デフォルト値です。
memstats		メモリー割り当てコードに関連した統計情報カウンターを表示します。
monlist		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表示します。

項目
peers

説明

サーバーが状態を保守するピアの一覧を、その状態の要約とともに表示します。要約情報には、下記のものが含まれます。

- リモート・ピアのアドレス
- リファレンス ID (不明リファレンス ID の場合、0.0.0.0)
- リモート・ピアの層 (層が 16 の場合、リモート・ピアが同期されていないことを示します)
- ポーリング間隔 (秒)
- 到達可能レジスター (8 進数)
- ピアの現在の見積遅延、オフセット、および分散 (秒数)

左マージンの文字は、このピア・エントリーが入っているモードを示します。

- + シンメトリック・アクティブ。
- シンメトリック・パッシブ。
- = クライアント・モードでポーリングされたリモート・サーバー。
- ^ サーバーは、このアドレスにブロードキャスト中。
- ~ リモート・ピアは、ブロードキャストを送信中。
- * サーバーが現在同期中のピアをマーク。

ホスト・フィールドの内容は、ホスト名でも、IP アドレスでも、そのパラメーターまたは REFCLK (インプリメンテーション番号、パラメーター) を持つリファレンス時計インプリメンテーション名でもかまいません。**hostnames no** の使用中は、IP アドレスのみが表示されます。

pstats PeerAddress [Addr2] [Addr3] [Addr4]

指定したピアに関連したピアごとの統計情報カウンターを表示します。

reslist

制約事項の適用方法を理解するのに役立つ可能性があるサーバーの制約事項リストを表示します。

sysinfo

ローカル・サーバーに関連したさまざまなシステム状態変数を表示します。最後の 4 行を除くすべての行が NTP バージョン 3 仕様、RFC 1305 で説明されています。システム・フラグは、さまざまなシステム・フラグを表示し、そのいくつかは、**enable** および **disable** 構成ステートメントによって設定し、消去することができます。**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す。これは、保守およびデバッグに使用します。ほとんどのアーキテクチャーでは、最初、この値は 500 ppm の大きな値から .01 から 0.1 ppm までの範囲の公称値に減少します。その値がデーモンの始動後もしばらくの間大きいままであれば、ローカル時計に異常があるか、カーネル変数の値 *Tick* を誤った可能性があります。**broadcastdelay** は、**broadcastdelay** 構成ステートメントによって設定されたとおりのデフォルトのブロードキャストの遅延を示し、**authdelay** は、**authdelay** 構成ステートメントによって設定されたとおりのデフォルトの認証の遅延を示します。

sysstats

プロトコル・モジュールで保守される統計情報カウンターを表示します。

timerstats

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情報カウンターを表示します。

ランタイム構成要求サブコマンド

サーバーは、構成された NTP キーを使用して、サーバーの状態変更発生を引き起こすすべての要求を認証します。また、サーバーは、キーを構成せずにこの機能を使用不可にすることもできます。**xtnpdc** コマンドが認識できるキー番号および対応キーを作成する必要があります。これを実行するには、暗号キーとして使用するパスワードを端末で要求する **keyid** サブコマンドおよび **passwd** サブコマンドを使用します。また、**xtnpdc** コマンドは、結果的にサーバーへ認証要求するサブコマンドを最初に出したときにキー番号およびパスワードの両方を自動的に要求します。認証によって、リクエスターが変更を行うことについてアクセス権があることを検査するだけでなく、伝送エラーからも保護します。

認証要求は、認証コードの計算と同じようにパケット・データに常にタイム・スタンプを取り込みます。サーバーは、このタイム・スタンプとパケットの受け取り時間とを比較します。

その時間差が 10 秒を超える場合、サーバーは要求を拒否します。これによって、LAN 上で通信内容を盗み聞きできる者によるサーバーへの単純な繰り返しのアタックをかなり困難にします。また、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もより困難にします。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から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提供すれば、実行時の再構成機能によってセキュリティーの水準が十分に確保されるはずです。

下記のサブコマンドは、どれも認証要求を実行します。

項目	説明
addpeer <i>PeerAddress</i> [<i>Keyid</i>] [<i>Version</i>] [prefer]	シンメトリック・アクティブ・モードで操作中の構成されたピアの関連付けを指定のアドレスに追加します。このサブコマンドを使用する場合、同じピアの既存の関連付けを削除するか、既存の関連付けを切り替えて、新規の構成に適合させることができます。 <i>Keyid</i> がゼロ以外の整数である場合、リモート・サーバーへのすべての出力パケットに、このキーが暗号化された認証フィールドが付加されます。認証を指定しない場合、 <i>Keyid</i> を 0 として入力するか、ブランクのままにします。 <i>Version</i> の値として 1、2、または 3 を取ることができ、3 がデフォルト値です。 prefer オプションは、可能な場合、主に時計の同期に使用される優先ピアを示します。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切である場合、PPS 信号にもそれが当てはまりません。
addserver <i>PeerAddress</i> [<i>Keyid</i>] [<i>Version</i>] [prefer]	操作モードがクライアントであることを除いて、 addpeer サブコマンドと同じです。
addtrap <i>Address</i> [<i>Port</i>] [<i>Interface</i>]	指定のローカル・インターフェース・アドレスを持つメッセージを送信するために、指定のアドレスおよびポート番号で非同期のメッセージに対するトラップをクリアします。ポート番号を指定しなければ、デフォルトで値が 18447 になります。インターフェース・アドレスを指定しなければ、デフォルトで値がローカル・インターフェースの送信元アドレスになります。
authinfo	認識されたキーおよび実行された暗号化および暗号解除のカウントを含む認証モジュールに関する情報を表示します。
broadcast <i>PeerAddress</i> [<i>Keyid</i>] [<i>Version</i>]	操作モードがブロードキャストであることを除いて、 addpeer サブコマンドと同じです。 <i>PeerAddress</i> 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 (224.0.1.1) に割り当てられたマルチキャスト・グループ・アドレスである可能性があります。
clrtrap <i>Address</i> [<i>Port</i>] [<i>Interface</i>]	指定のアドレスの非同期メッセージ用トラップおよびメッセージ送信用ポート番号を指定のローカル・インターフェース・アドレスによって消去します。ポート番号で非同期のメッセージに対するトラップをクリアします。ポート番号を指定しなければ、デフォルトで値が 18447 になります。インターフェース・アドレスを指定しなければ、デフォルトで値がローカル・インターフェースの送信元アドレスになります。
delrestrict <i>Address Mask</i> [ntpport]	制約事項リストからマッチング・エントリーを削除します。
disable <i>Option</i> ...	さまざまなサーバー・オプションを使用不可にします。記述されていないオプションには影響しません。 enable サブコマンドは、オプションを説明します。

項目	説明
enable <i>Option ...</i>	さまざまなサーバー・オプションを使用可能にします。記述されていないオプションには影響しません。下記の <i>Option</i> の値を 1 つ以上指定することができます。
auth	トラステッド鍵およびキー ID を使用して、ピアが正しく認証された場合にのみ、構成解除されたピアにサーバーを同期させます。デフォルトでは、このオプションは、使用不可 (オフ) です。
bclient	ブロードキャスト・サーバーまたはマルチキャスト・サーバーからサーバーにメッセージを listen させ、次にそのサーバーに対する関連付けを自動的にインスタンス化します。デフォルトでは、この引数は、使用不可 (オフ) です。
monitor	デフォルトの使用可能 (オン) によってモニター機能を使用可能にします。
pll	デフォルトの使用可能 (オン) によってサーバーがそのローカル時計をアジャストできるようにします。設定されなければ、ローカル時計は組み込み時間および頻度オフセットで自由に実行されます。このオプションは、ローカル時計がその他のデバイスまたはプロトコルによってコントロールされ、NTP が他のクライアントに同期を提供するためにのみ使用されている場合に便利です。
fudge <i>PeerAddress</i> [<i>Time1</i>] [<i>Time2</i>] [<i>Stratum</i>] [<i>Refid</i>]	<p>stats デフォルトの使用可能 (オン) によって統計情報機能 filegen を使用可能にします。リファレンス時計に対して一定のデータを設定する方法を提供します。</p> <p><i>Time1</i> および <i>Time2</i> は、固定小数点秒数で示されており、いくつかの時計ドライバーで調整定数として使用されます。</p> <p><i>Stratum</i> は、0 から 15 までの範囲の数であり、標準外操作層を時計に割り当てる場合に使用します。</p> <p><i>Refid</i> は、1 から 4 文字までの範囲の ASCII 文字列で、標準外リファレンス ID を時刻に割り当てる場合に使用します。</p>
monitor yes no	モニター機能を使用可能または使用不可にします。 monitor no サブコマンドに monitor yes サブコマンドを後続させるのは、パケット・カウントをリセットする方法として適切です。
readkeys	認証キーの現在の設定を除去し、 xntpd 構成ファイルに指定されたキー・ファイルを再読み取りして、新規の設定を取得します。これによって、サーバーを再始動せずに暗号キーを変更することが許可されます。
reset <i>Module</i>	サーバーのさまざまなモジュールにおける統計情報カウンターを消去します。 io 、 sys 、 mem 、 timer 、 auth 、 allpeers の <i>Module</i> の値を 1 つ以上指定することができます。

項目	説明
restrict <i>Address Mask Option ...</i>	<i>Option</i> の値を既存の制約事項リスト・エンタリーに追加するか、指定の <i>Option</i> によって新規の入力を一覧に追加します。 mask オプションは、デフォルトで 255.255.255.255 になります。これは、 <i>Address</i> が個別ホストのアドレスとして扱われることを意味します。下記の <i>Option</i> の値を 1 つ以上指定することができます。
ignore	このエンタリーにマッチするホストからのすべてのパケットを無視します。照会にもタイムサーバー・ポーリングにも応答しません。
limited	これらのホストが同じ net のクライアント制約事項に従うことを指定します。このコンテキストにおける net は、net の IP 概念 (クラス A、クラス B、クラス C など) を参照します。最後の client_limit_period 秒間アクティブであった最初の client_limit ホストのみを受け入れます。同じ net のその他のクライアントからの要求を拒否します。時間要求パケットのみを考慮に入れます。プライベート・パケット、コントロール・パケット、およびブロードキャスト・パケットはクライアント制約事項に従わないため、クライアント・カウントには加算されません。 xntpd デーモンのモニター機能によって、クライアントのヒストリーが保持されます。このオプションを使用する場合、モニターはアクティブなままにしておきます。 client_limit のデフォルト値は 3 です。 client_limit_period のデフォルト値は 3600 秒です。
lowpriotrap	ホストを低優先順位状況へマッチさせて設定したトラップを宣言します。サーバーは、先着順サービスによって割り当てられた、限られた数のトラップ (現在の制約事項は 3) を保守することができ、その後のトラップ要求についてはサービスを拒否します。このパラメーターは、後で要求される通常優先順位のトラップが低い優先順位のトラップを上書きすることを許可して、割り当てアルゴリズムを変更します。
nomodify	サーバーの状態 (実行時の再構成) を変更しようとするすべての NTP モードの 6 および 7 パケットを無視します。情報を戻す照会を許可します。
nopeer	ホストへのポーリング要求に対してステートレス・タイム・サービスを提供しますが、これらのホストにピア・メモリー・リソースを割り当てることはしません。
noquery	ソースからのすべての NTP モードの 6 および 7 パケット (情報照会および構成要求) を無視します。タイム・サービスには影響しません。
noserve	モードが 6 または 7 ではない NTP パケットを無視します。これによってタイム・サービスが拒否されますが、照会は許可されます。
notrap	モード 6 コントロール・メッセージ・トラップ・サービスをマッチング・ホストに提供することを拒否します。トラップ・サービスは、リモート・イベント・ログ・プログラム用のモード 6 コントロール・メッセージ・プロトコルのサブシステムです。
notrust	その他の関連付けでは、これらのホストを通常どおり扱いますが、同期化ソースとして使用することは絶対にありません。
ntpport	パケットのソース・ポートが標準 NTP UDP ポート (123) である場合にのみ制約事項エンタリーをマッチさせます。
setprecision <i>Precision</i>	サーバーが公示する精度を設定します。 <i>Precision</i> は、-4 から -20 までの範囲の負の整数である必要があります。
traps	サーバーに設定したトラップを表示します。
trustkey <i>Keyid ...</i>	トラステッド鍵リストに 1 つ以上のキーを追加します。認証を使用可能にする場合、トラステッド鍵を使用してトラステッド時間によるピアを認証します。
unconfig <i>PeerAddress</i> [<i>Addr2</i>] [<i>Addr3</i>] [<i>Addr4</i>]	指定したピアから構成されたビットを除去します。大抵、ピアの関連付けを削除します。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、関連付けを構成解除されたモードで持続させてもかまいません。
unrestrict <i>Address Mask Option ...</i>	<i>Address</i> および <i>Mask</i> によって示された制約事項リスト・エンタリーから指定のオプションを除去します。 restrict サブコマンドは、 <i>Option</i> の値を説明します。
untrustkey <i>Keyid ...</i>	トラステッド鍵リストから 1 つ以上のキーを除去します。

終了状況

このコマンドは、下記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。

セキュリティー

アクセス制御: このコマンドを実行するには、システム・グループの一部である必要があります。

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。

監査イベント: N/A

指定したピアに関連したピアごとの統計情報カウンターを表示します。

例

1. Network Time Protocol デモンから照会/コントロール・プログラムを始動するには、下記を入力します。

```
xntpd
```

2. ホスト 9.3.149.107 上のアドレス 127.127.1.0 にピアの統計情報カウンターを表示するには、下記を入力します。

```
xntpd -c "pstats 127.127.1.0" 9.3.149.107
```

下記のような出力が表示されます。

```
remote host: LOCAL(0)
local interface: 127.0.0.1
time last received: 49s
time until next send: 15s
reachability change: 818s
packets sent: 13
packets received: 13
bad authentication: 0
bogus origin: 0
duplicate: 0
bad dispersion: 4
bad reference time: 0
candidate order: 1
```

ファイル

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

関連情報:

`ntp` コマンド

213 ページの『`xntpd` デーモン』

RBAC

Trusted AIX

xpr コマンド

目的

プリンターへの出力のために、ウィンドウ・ダンプ・ファイルをフォーマットします。

構文

```
xpr [ -append FileName [ -noff ] | -output FileName ] [ -landscape | -portrait ] [ -compact ]
[ -cutoff Level ] [ -density Dpi ] [ -gray { 2 | 3 | 4 } ] [ -header String ] [ -height Inches ]
[ -left Inches ] [ -noposition ] [ -plane PlaneNumber ] [ -psfig ] [ -report ] [ -rv ] [
-scale Scale ] [ -split Number ] [ -top Inches ] [ -trailer String ] [ -width Inches ] [
-device Device ] [ ImageFile ]
```

説明

`xpr` コマンドは、`xwd` ユーティリティーによって生成されたウィンドウ・ダンプ・ファイルを入力として使用し、このダンプ・ファイルをハードウェアでサポートされているすべてのプリンターに出力するためにフォーマットします。ファイル引数を指定しなければ、`xpr` コマンドは標準入力を使用します。デフォルトでは `xpr` コマンドは出力ページ上に可能な限り大きなウィンドウを出力します。

`xpr` コマンドのオプションを使用すると、ヘッダーやトレーラーを追加し、マージンを指定し、スケールや方向を調整し、複数のウィンドウ・ダンプを単一の出力ファイルに追加することができます。出力は、`-output` フラグが指定されていない限り標準出力です。

フラグ

項目	説明
<code>-append FileName</code>	ウィンドウを追加するファイル (以前に <code>xpr</code> コマンドによって作成されている) の名前を指定します。(このフラグは PostScript プリンターではサポートされていません。)
<code>-compact</code>	ホワイト・ピクセルが多いウィンドウを小さく表示するために、簡単なラン・レンジス・エンコード法を使います。このフラグを指定するとホワイト・スペースは圧縮されますが、ブラック・スペースは圧縮されないで、反転表示ウィンドウでは有効ではありません。
<code>-cutoff Level</code>	(このフラグは PostScript プリンター、LIPS II+、LIPSIII 出力でのみサポートされます)。LaserJet プリンター上でモノクローム出力用にカラーが黒または白にマップされる際の輝度を変更します。 <code>Level</code> 変数は、全輝度に対するパーセントで表されます。小数部も使用できません。

項目	説明
-device <i>Device</i>	ファイルが印刷されるデバイスを指定します。 xpr コマンドでは下記のプリンターがサポートされています。
	3812 または pp IBM PP3812
	4207 Proprinter
	5201 IBM Quietwriter 1 モデル 2
	5202 IBM Quietwriter 2
	jprinter IBM 日本語プリンター (日本語データ・ストリーム)
	ljet HP LaserJet および IBM Laser Printer
	ps PostScript プリンター (デフォルト)
	lips2 Canon LaserShot LIPS II+ モード
	lips3 Canon LaserShot LIPS III モード
-density <i>Dpi</i>	HP プリンターで使用する 1 インチ当たりのドット (dpi) 密度を指定します。デフォルト値は 300 dpi です。300、150、100、75 dpi が指定できます。
-gray <i>Number</i>	白黒イメージへのマッピングではなく、カラー・イメージへのグレースケール変換を指定します。 <i>Number</i> 変数に、下記のいずれかの値を指定しなければなりません。
	2 2 x 2 の変換
	3 3 x 3 の変換
	4 4 x 4 の変換
	これにより、変換時のイメージの有効幅および高さが、それぞれ 2 倍、3 倍および 4 倍になります。
	注: このオプションは、PostScript プリンターにのみ有効です。
-header <i>String</i>	ウィンドウの上に表示されるヘッダー文字列を指定します。
-height <i>Inches</i>	1 ページの中のウィンドウの高さの最大値を指定します。
<i>ImageFile</i>	捕そくしたイメージのビットマップが入ったファイルです。 <i>ImageFile</i> パラメーターを指定しなければ、 xpr コマンドは標準入力を使用します。
-landscape	ウィンドウをランドスケープ・モードで出力します。(ディスプレイでは、ウィンドウを縦よりも横の方を長くしています。) デフォルトではウィンドウは、長い方の辺が用紙の長い方の辺に出力されます。
-left <i>Inches</i>	左マージンをインチで指定します。小数部も使用できます。デフォルトでは、このフラグを指定すると、ウィンドウはページの中央に出力されます。
-noff	-append フラグと一緒に指定すると、ウィンドウが以前のウィンドウと同じページに表示されます。(このフラグは PostScript プリンターではサポートされていません。)
-noposition	LaserJet プリンター用に、ヘッダー、トレーラー、およびイメージの位置決めコマンドの生成をバイパスさせます。
-output <i>FileName</i>	出力ファイル名を指定します。このオプションを指定しなければ、 xpr コマンドは標準出力を使用します。
-plane <i>PlaneNumber</i>	どのビット・プレーンをイメージ内で使用するかを指定します。デフォルトでは、イメージ全体が使用され、カラー輝度に基づいて値が黒および白にマップされます。このオプションは、LaserJet プリンターではサポートされません。
-portrait	ウィンドウをポートレイト・モードで出力します。(ディスプレイでは、ウィンドウを横よりも縦の方を長くしています。) デフォルトではウィンドウは、長い方の辺が用紙の長い方の辺に出力されます。
-psfig	PostScript ピクチャーがページの中央に変換されないようにします。
-report	ウィンドウの <i>ImageFile</i> パラメーターについての統計情報を標準エラーに出力します。
-rv	ウィンドウを反転表示で出力します。

項目	説明
-scale <i>Scale</i>	ページ上のウィンドウのサイズを変更します。PostScript プリンターでは、ウィンドウ・ピクスマップの各ビットを、指定されたサイズのグリッドへ変換できます。例えば、各ビットを 3 × 3 のグリッドに変換できます。3 X 3 のグリッドを指定するには、 -scale 3 と入力します。デフォルトではウィンドウは、指定された方向にページに納まる最大の大きさで出力されます。縦横比は、デバイスが指定されていなければ変更できます。
-split <i>Number</i>	ウィンドウを複数ページに分割して出力します。このファンクションは、ウィンドウが大きすぎてプリンターに過剰な負荷をかけ、ページの出力を正常に行えない場合に必要となります。(このフラグは PostScript プリンターや HP Laserjet プリンターではサポートされていません。)
-top <i>Inches</i>	ウィンドウの上マージンをインチで指定します。小数部も使用できます。デフォルトでは、このフラグを指定すると、ウィンドウはページの中央に出力されます。
-trailer <i>String</i>	ウィンドウの下に表示されるトレーラー文字列を指定します。
-width <i>Inches</i>	ページの最大幅を指定します。

注: 4207, 5201, 5202 プリンターのイメージは、**xwd** ユーティリティを使って、XYPixmap または XYBitmap フォーマットで記録しなければなりません。XYPixmap イメージは、しきい値アルゴリズムを使ってビットマップに変換されます。HP Laserjet プリンターの場合、複数面イメージは ZPixmap フォーマットで記録されなければなりません。単一面イメージは、XYPixmap、XYBitmap、ZPixmap のいずれのフォーマットでもかまいません。

関連資料:

- 269 ページの『xwd コマンド』
- 270 ページの『xwud コマンド』

xpreview コマンド

目的

X ディスプレイ上に troff ファイルを表示します。

構文

```
xpreview [ -BackingStore BackingStoreType ] [ -page Number ] [ ToolKitFlag ... ] { File | - }
```

説明

xpreview は、AIXwindows ディスプレイ上に **troff** コマンドからの出力を表示する AIXwindows 2.1 および Motif2.1 ベースのアプリケーションです。 **troff** コマンド出力ファイルは、devX100 デバイス用に準備しておかなければなりません。

ユーザー・インターフェースには、ルート・メニューの呼び出し、ウィンドウのアイコン化、およびウィンドウのフルスクリーン・サイズ設定などを制御する標準 AIXwindows インターフェースが含まれます。インターフェースには、また、テキスト用のスクロール可能なディスプレイ領域を持つメイン・ウィンドウも含まれます。Next、Previous、Goto Page、Print Page、Print File、Newfile のプッシュボタンを使って、文書の表示を操作します。

マウス・ボタン 3 を使うと、印刷機能を構成するためのポップアップ・メニューを使えるようになります。メニューには、コマンド・ラインを設定するオプションや、プリンター・キューを選択するオプションが含まれています。コマンド・ライン・ダイアログ・ボックスは、**troff** コマンドを通しての、コマンド・ライン入力用です。例えば、以下のとおりです。

```
pic -Tibm3816 troff-input-file |tbl|troff -mm -Tibm3816
```

コマンド・ラインとして受け付けられます。プリンター・キュー・オプションは、構成されたプリンター・キューの一覧を表示します。このオプションが選択されていない場合は、**xpreview** コマンドは、システムの定義したデフォルトのキューを使用します。

入力ファイルをプレビューしているときは、**Print Page** および **Print File** ボタンは、コマンド・ライン入力を要求します。いったんプリンター・キューが選択されると、表示セッションの間、代わりにプリンター・キューが選択されるまでずっと以前のキューがそのまま残ることに注意してください。

ヨーロッパ・ロケールでは、**devX100** デバイス向けに下記のフォントがサポートされています。

- Times New Roman の標準、イタリック、および太字
- Courier の標準および太字
- Helvetica の標準および太字
- 記号

xpreview コマンドでサポートされているフォント・サイズは 8、10、14、18、24、30、36 ポイントです。

xpreview コマンドは、本文中に説明されているもの以外のデバイス用に構成された **troff** コマンドによるファイル出力結果は表示しません。

ある特定のデバイス上でファイルをプレビューする際、**xpreview** コマンドでは、下記のディレクトリー内にあるフォントが必要になります。

- 日本語以外のフォント・ファイル向けにフォーマットされているファイルは、**/usr/lib/X11/fonts** ディレクトリー。
- 日本語フォント・ファイルは、**/usr/lib/X11/fonts/JP** ディレクトリー。

マルチバイトのサポート

xpreview コマンドでは、マルチバイトのロケールがサポートされます。また、日本語文字を表示するには、日本語 16 ドット・フォント (日本語 BSL パッケージの一部) と 24 および 32 ドット・フォント (AIXwindows フォント・パッケージの一部) をインストールしておかなければなりません。韓国語文字を表示するには、韓国語フォント (韓国語 BSL パッケージの一部) をインストールしておかなければなりません。

現在サポートされている日本語には、下記のフォント・セットがあります。

- 16 ドット: RomanKn12、Kanji12、および IBM_JPN12
- 24 ドット : RomanKn17、Kanji17、および IBM_JPN17
- 32 ドット : RomanKn23、Kanji23、および IBM_JPN23、または RomanKn23G、Kanji23G、および IBM_JPN23G

現在サポートされている韓国語には、下記のフォント・セットがあります。

- 16 ドット: EnglHg16 および Hangul16
- 24 ドット : EnglHg24 および Hangul24

フラグ

xpreview コマンドでは、標準 **X Toolkit** コマンド・ライン・フラグを、下記に示すフラグと同様に使うことができます。

項目		説明
-		標準入力から読み取られる入力を要求します。
-help		使用できるコマンド・ライン・フラグの簡潔な要約が出力されることを示します。
-BackingStore	<i>BackingStoreType</i>	<p>-BackingStore フラグを指定すると、サーバーはウィンドウの内容を保管します。ビューポートの中でスクロールされたときに、サーバー・バッキング・ストアに保管された内容がウィンドウに描画されます。ウィンドウの再描画は 1 秒ぐらいかかることがあります。<i>BackingStoreType</i> パラメーターは、Always、WhenMapped、NotUseful のいずれかの値を取ります。</p> <p>ヒント: -BackingStore フラグと <i>BackingStoreType</i> パラメーターの間にはスペースを入力してください。</p> <p>要件: このフラグを用いるには、サーバーがバッキング・ストア可能な状態で始動している必要があります。</p> <p>最初に表示される文書のページ番号を指定します。</p> <p>下記の標準 X Toolkit フラグは普通 xpreview コマンドと一緒に使われます。</p>
-page	<i>Number</i>	
ToolkitFlag		
		<p>-bg <i>Color</i></p> <p>ウィンドウのバックグラウンドに使用する色を指定します。デフォルトは白です。</p> <p>-bg <i>Color</i></p> <p>ウィンドウのバックグラウンドに使用する色を指定します。デフォルトは白です。</p> <p>-fg <i>Color</i></p> <p>テキストの表示に使用する色を指定します。デフォルトは黒です。</p> <p>-geometry<i>Geometry</i></p> <p>ウィンドウに適したサイズと位置を指定します。</p> <p>-display <i>Host:Display</i></p> <p>接続する X サーバーを指定します。</p> <p>-xrm<i>ResourceString</i></p> <p>使用するリソース文字列を指定します。</p> <p>出力されるファイルを指定します。</p>
ファイル		

例

1. **troff** コマンドから出力されたファイルを **xpreview** コマンドで使用できるファイルとして作成するには、下記のコマンドを入力します。

```
troff-TX100 troff-input | xpreview
pic -TX100 pic-troff-input | tbl | troff -man -TX100 | xpreview
```

2. **troff** コマンドから出力されたファイルを日本語バージョンの **xpreview** コマンドで使用できるファイルとして作成するには、下記のコマンドを入力します。

```
LANG=ja_JP
troff -TX100 troff-input | xpreview -
pic -TX100 pic-troff-input | tbl | troff -man -TX100 ¥
| xpreview -
```

ファイル

項目	説明
<code>/usr/lib/X11/app-defaults/XPreview</code>	ユーザーが構成可能なアプリケーション・デフォルト・ファイルが入っています。
<code>/usr/lib/X11/Ja_JP/app-defaults/XPreview</code>	日本語 (IBM-943) ロケール用にユーザーが構成可能なアプリケーション・デフォルト・ファイルが入っています。
<code>/usr/lib/X11/ja_JP/app-defaults/XPreview</code>	日本語 (IBM-eucJP) ロケール用にユーザーが構成可能なアプリケーション・デフォルト・ファイルが入っています。
<code>/usr/lib/X11/ko_KR/app-defaults/XPreview</code>	韓国語ロケール用にユーザーが構成可能なアプリケーション・デフォルト・ファイルが入っています。
<code>/usr/lib/X11/zh_TW/app-defaults/XPreview</code>	中国語 (繁体字) ロケール用にユーザーが構成可能なアプリケーション・デフォルト・ファイルが入っています。
<code>/usr/lib/font/devX100</code>	devX100 デバイス用の troff フォントが入っています。
<code>/usr/lib/X11/fonts</code>	100 dpi デバイス用の X フォントが入っています。
<code>/usr/lib/X11/fonts/JP</code>	マルチバイト文字用の X フォントが入っています。
<code>/usr/lib/X11/fonts/JP</code>	日本語文字用の X フォントが入っています。

関連情報:

troff コマンド

xprofiler コマンド

目的

GUI ベースの AIX パフォーマンス・プロファイル作成ツールである Xprofiler を起動します。

構文

```
xprofiler [ program ] [ -b ] [ -s ] [ -z ] [ -a path ] [ -c file ] [ -L pathname ] [ [ -e function ] ... ] [ [ -E function ] ... ] [ [ -f function ] ... ] [ [ -F function ] ... ] [ -disp_max number_of_functions ] [ [ gmon.out ] ... ]
```

`xprofiler -h` | `-help`

説明

`xprofiler` コマンドは、GUI ベースの AIX パフォーマンス・プロファイル作成ツールである Xprofiler を起動します。Xprofiler は、シリアル・アプリケーションおよびパラレル・アプリケーションのパフォーマンスの分析に使用します。Xprofiler は `-pg` コンパイル・オプションで収集されたデータを使用し、いくつかのレポート・ウィンドウでテキスト・データを表示するだけでなく、アプリケーションの関数をグラフィカル表現で示します。この表示フォーマットは、最もプロセッサに集中する関数を識別することを意図しています。

フラグ

項目	説明
-a	ライブラリー・ファイルおよびソース・コード・ファイルの代替検索パスを指定します。複数のパスを指定する場合はパスを「 」で囲み、各パスを「:」かスペースで区切る必要があります。
-b	フラット・プロファイル、コール・グラフ・プロファイル、および関数索引のレポートが、「File (ファイル)」メニューの「Save As (別名保存)」オプションでファイルに書き込まれるときに、それらのレポートのフィールド説明が出力されないようにします。
-c	起動された Xprofiler でどの関数を表示するかを決定するための情報が含まれる構成ファイルをロードします。
-disp_max	Xprofiler が関数呼び出しツリーに最初に表示する関数ボックスの数を設定します。このフラグで指定する値は、0 から 5,000 までの任意の整数です。Xprofiler は、最もプロセッサに集中する関数の関数ボックスを、指定した数だけ表示します。例えば 50 を指定すると、Xprofiler は、プログラムの中で最もプロセッサを使用する 50 の関数の関数ボックスを表示します。この後、表示される関数ボックスの数を、「Filter (フィルター)」メニュー・オプションで変更することができます。このフラグは Xprofiler のレポートの内容には影響しません。
-e	関数呼び出しツリー内の指定した関数の関数ボックスの外観の強調表示を解除し、コール・グラフ・プロファイル・レポートでのその関数のエントリーの数を制限します。これは、指定関数の子孫が非指定関数によって呼び出されない限り、指定関数の子孫にも適用されます。指定関数の関数ボックスは、関数呼び出しツリーでは、グレイ表示になります。そのサイズとラベルの内容は、変わりません。これは、子孫関数が非指定関数によって呼び出されない限り、子孫関数にも適用されます。指定関数のエントリーがコール・グラフ・プロファイル・レポートに表示されるのは、別の関数の子となる場所に表示されるか、少なくとも 1 つの非指定関数も親として持つ関数の親として表示される場合だけです。このエントリーの情報は変わりません。指定関数の子孫のエントリーは、指定関数の子孫がプログラム内の少なくとも 1 つの非指定関数によって呼び出されない限り、表示されません。
-E	関数呼び出しツリー内の指定した関数の関数ボックスの外観およびラベル情報を変更します。コール・グラフ・プロファイル・レポートでのこの関数のエントリーの数の制限、および関連するプロセッサ・データの変更も行います。この結果は、指定関数の子孫がプログラム内の非指定関数によって呼び出されない限り、指定関数の子孫にも適用されます。指定関数の関数ボックスは、関数呼び出しツリーではグレイ表示され、そのサイズと形状も変わって最小許容サイズの正方形で表示されるようになります。さらに、関数ボックス・ラベルに表示されるプロセッサ時間も、0 (ゼロ) と表示されます。これは、子孫関数が非指定関数によって呼び出されない限り、子孫関数の関数ボックスにも同様に適用されます。このオプションを指定すると、指定関数の上位ごとの関数ボックスのラベルの左側のプロセッサ合計から、指定関数が使用したプロセッサ時間が差し引かれるようになります。指定関数のエントリーがコール・グラフ・プロファイル・レポートに表示されるのは、別の関数の子となる場所に表示されるか、少なくとも 1 つの非指定関数も親として持つ関数の親として表示される場合だけです。この場合、このエントリーの自身および子孫の列に示される時間は、0 (ゼロ) に設定されます。さらに、指定関数の子孫列にあった合計時間が、プロファイル作成される関数の子孫列に示される時間から差し引かれます。その結果、このレポートでプロファイル作成される関数の % 時間列に示される値が変わるということに注意してください。
-f	関数呼び出しツリー内の、指定した関数とその子孫以外の、すべての関数ボックスの外観の強調表示を解除します。さらに、コール・グラフ・プロファイル・レポートでの、非指定関数と非子孫関数のエントリーの数が制限されます。 -f フラグは -e フラグをオーバーライドします。指定関数とその子孫以外のすべての関数ボックスは、関数呼び出しツリーではグレイ表示されます。このボックスのサイズとラベルの内容は、変わりません。指定関数とその子孫の関数ボックスとラベルの外観に変化はありません。非指定または非子孫関数のエントリーがコール・グラフ・プロファイル・レポートに表示されるのは、指定関数の親または子となる場所に表示されるか、指定関数の子孫の 1 つとなる場所に表示される場合のみです。このエントリーのすべての情報は変わりません。
-F	関数呼び出しツリー内の、指定した関数とその子孫以外の、すべての関数ボックスの外観およびラベル情報を変更します。さらに、コール・グラフ・プロファイル・レポートでの、非指定関数と非子孫関数のエントリーの数が制限され、非指定関数と非子孫関数に関連したプロセッサ・データが変更されます。-F フラグは -E フラグをオーバーライドします。指定関数の関数ボックスは、関数呼び出しツリーではグレイ表示され、そのサイズと形状も変わって最小許容サイズの正方形で表示されるようになります。さらに、関数ボックス・ラベルに表示されるプロセッサ時間も、0 (ゼロ) と表示されます。非指定または非子孫関数のエントリーがコール・グラフ・プロファイル・レポートに表示されるのは、指定関数の親または子となる場所に表示されるか、指定関数の子孫の 1 つとなる場所に表示される場合のみです。このエントリーの自身および子孫の列に示される時間は、0 (ゼロ) に設定されます。この場合、このエントリーの自身および子孫の列に示される時間は、0 (ゼロ) に設定されます。その結果、このレポートでプロファイル作成される関数の % 時間列に示される値が変わるということに注意してください。
-h -help	Xprofiler の使用方法を STDERR に書き込んでから終了します。この情報には、xprofiler コマンド・ラインの構文、および Xprofiler ランタイム・オプションの説明が含まれます。
-L	共用ライブラリーの検索に代替パス名を使用します。複数のパスを指定する場合は、Xprofiler GUI の「File (ファイル)」メニューの「Set File Search Paths (ファイル検索パスの設定)」オプションを使用してください。
-s	Xprofiler の始動時に複数の gmon.out ファイルが指定された場合に、gmon.sum プロファイル・データ・ファイルを作成します。gmon.sum ファイルは、指定されたすべてのプロファイル・ファイルに含まれるプロファイル情報の和になります。単一の gmon.out ファイルを指定した場合の gmon.sum ファイルには、その gmon.out ファイルと同じデータが含まれるので注意してください。

項目	説明
-z	プロセッサをまったく使用しておらずコール・カウントもない関数を、フラット・プロファイル、コール・グラフ・プロファイル、および関数索引のレポートに含めます。自身の定義を含むファイルが -pg オプションを指定してコンパイルされていない場合 (システム・ライブラリー・ファイルでは普通のことです)、関数のコール・カウントはありません。

例

xprofiler を使用するには、まずプログラム (例えば **foo.c**) を、次のように **-pg** を指定してコンパイルする必要があります。

```
xlc -pg -o foo foo.c
```

1. プログラム **foo** が実行されると、その実行に関連するプロセッサごとに **gmon.out** ファイルが 1 つ生成されます。**xprofiler** を起動するには、次のように入力します。

```
xprofiler foo [[gmon.out]...]
```

ファイル

項目	説明
<code>/usr/lib/X11/app-defaults/Xprofiler</code>	xprofiler コマンドのロケーション。

xrdb コマンド

目的

X サーバーのリソース・データベース・ユーティリティを実行します。

構文

```
xrdb [ -display Display ] [ -help ] [ -quiet ] [ -retain ] [ -cpp FileName | -nocpp ] [ -D Name=Value ] [ -I Directory ] [ -U Name ] [ -all | -global | -screen | -screens ] [ -n ] [ -edit FileName | [ -backup String ] | -merge [ FileName ] | -load [ FileName ] | -query | -remove | symbols ] -override ]
```

説明

xrdb コマンドは、画面 0 のルート・ウィンドウにある `RESOURCE_MANAGER` 属性またはフルスクリーン・ウィンドウにある `SCREEN_RESOURCES` 属性、あるいはその組み合わせのすべての内容を取得または設定します。このプログラムは通常、X 始動ファイルから実行します。

ほとんどの X クライアントは、`RESOURCE_MANAGER` および `SCREEN_RESOURCES` 属性を使って、アプリケーションのための色やフォントなどをユーザーの好みに合わせて設定するための情報を取り出します。これまでの X のバージョンでは使用するコンピューターのすべてで デフォルト・ファイルを保守しなければなりませんでした。この情報をディスク上ではなく (すべてのクライアントが使用できる) サーバー上に持つことによって、この問題を解決しています。またファイルを編集せずにデフォルトを動的に変更することもできます。

`RESOURCE_MANAGER` 属性は、ディスプレイのすべての画面に適応するリソースを指定します。各スクリーン上の `SCREEN_RESOURCES` 属性は、その画面に使用される追加 (もしくは上書きする) リソースを指定します。(1 画面しかないときは、`SCREEN_RESOURCES` は、通常使用されません。すべてのリソースが `RESOURCE_MANAGER` 属性の中に置かれているのみです。)

互換性に関しては、(xrdp コマンドが実行されなかったか、または属性が除去されたという理由で) RESOURCE_MANAGER 属性が定義されていない場合、リソース・マネージャーは、ユーザーのホーム・ディレクトリーの中の **.Xdefaults** という名前のファイルを探します。

ファイル名 (- が指定されている場合、または入力ファイルが指定されていない場合は標準入力) は、使用されるサーバーの機能に基づき、定義された下記のシンボルを使って、オプションで C プリプロセッサーへ渡されます。

項目	説明
SERVERHOST= <i>Hostname</i>	接続しているディスプレイのホスト名の部分を指定します。
SRVR_ <i>name</i>	SERVERHOST ホスト名文字列をリーガル ID に変更します。例えば、my-dpy.lcs.mit.edu は、SRVR_my_dpy_lcs_mit_edu になります。
HOST= <i>Hostname</i>	接続しているディスプレイのホスト名の部分を指定します。
DISPLAY_NUM= <i>num</i>	サーバー・ホストのディスプレイ数を指定します。
CLIENTHOST= <i>Hostname</i>	xrdp が実行されているホスト名を指定します。
CLNT_ <i>name</i>	CLIENTHOST ホスト名文字列をリーガル ID に変更します。例えば、expo.lcs.mit.edu は、CLNT_expo_lcs_mit_edu になります。
WIDTH= <i>Number</i>	デフォルト画面の幅をピクセル単位で指定します。
HEIGHT= <i>Number</i>	デフォルト画面の高さをピクセル単位で指定します。
X_RESOLUTION= <i>Number</i>	デフォルト画面の X 軸レゾリューションをメートル当たりのピクセル数で指定します。
Y_RESOLUTION= <i>Number</i>	デフォルト画面の Y 軸レゾリューションをメートル当たりのピクセル数で指定します。
PLANES= <i>Number</i>	デフォルト画面のルート・ウィンドウのビット・プレーンの数 (奥行き) を指定します。
RELEASE= <i>Number</i>	サーバーのベンダー・リリース番号を指定します。この番号の解釈は、 VENDOR によって変化します。
REVISION= <i>Number</i>	サーバーによってサポートされている X プロトコルのマイナー・バージョンを指定します (現在は 0)。
VERSION= <i>Number</i>	サーバーによってサポートされている X プロトコルのメジャー・バージョンを指定します (常に 11 でなければなりません)。
VENDOR= <i>Vendor</i>	サーバーのベンダーを指定する文字列を指定します。
VNDR_ <i>name</i>	VENDOR 名文字列をリーガル ID に変更します。例えば、MIT X Consortium は、VNDR_MIT_X_Consortium になります。
EXT_ <i>name</i>	各拡張文字列をリーガル ID に変更します。サーバーがサポートする各プロトコル拡張に付きシンボルが定義されます。例えば、X3D-PEX は、EXT_X3D_PEX になります。
NUM_SCREEN= <i>num</i>	画面の合計数を指定します。
SCREEN_NUM= <i>num</i>	現在の画面の数を指定します。0 (ゼロ) から。
BITS_PER_RGB= <i>Number</i>	RGB カラー仕様に含まれる有効ビットの数を指定します。これは、ハードウェアが生成できる各 1 次局の異なる陰影数の底を 2 とした常用対数です。 PLANES とは無関係なので注意してください。
CLASS= <i>VisualClass</i>	デフォルト画面のルート・ウィンドウのビジュアル・クラスに、下記のいずれかを指定します。
CLASS_ <i>visualclass=visualid</i>	フォーム内のルート・ウィンドウのビジュアル・クラスに、 #ifdef をオンにできることを指定します。値は、ビジュアルの数値 id です。
CLASS_ <i>visualclass_depth=</i> <i>num</i>	DirectColor 、 GrayScale 、 PseudoColor 、 StaticColor 、 StaticGray 、 TrueColor スクリーンに対してサポートされている各ビジュアルにつきシンボルが定義されます。シンボルには、ビジュアル・クラスおよびその奥行きが含まれます。値は、ビジュアルの数値 id です。(同じクラスと奥行きのビジュアルが 2 つ以上ある場合、サーバーが報告する最初のビジュアルのサーバー数値 id が使用されます。)
COLOR	CLASS が、 StaticColor 、 PseudoColor 、 TrueColor 、または DirectColor の 1 つである場合にのみ定義されます。

注釈行は！(感嘆符)で始まり、無視されます。

xrdb を標準入力から読み取れるので、これを使って端末またはシェル・スクリプトから属性の内容を直接変更できます。

フラグ

項目	説明
-all	このオプションは、ディスプレイのすべての画面上で操作が画面固有属性 (SCREEN_RESOURCES) と同様に画面独立リソース属性 (RESOURCE_MANAGER) で行われることを示します。例えば、 -query と結び付けて使用されているときには、すべての属性の内容が出力されます。 -load および -merge の場合は、入力ファイルは、各画面につき一度ずつ処理されます。フルスクリーンの出力に共通して現れるリソースは、集められ、画面独立リソースとして適用されます。残りのリソースは、個別の画面ごとの属性に適用されます。これは、デフォルト・モードの操作です。このオプションは X11R5 に固有なものです。
-backup String	ファイル名に付けられる接尾部を指定します。 -edit フラグとともに使用して、バックアップ・ファイルを生成します。 -edit は、 -backup String の前提条件です。
-cpp FileName	使用する C プリプロセッサ・プログラムのパス名を指定します。 xrdb コマンドは CPP を使うように設計されていますが、フィルターとして動作し、 -D フラグ、 -I フラグと、 -U フラグを使用できるプログラムをすべて使うことができます。
-DName=Value	プリプロセッサへ渡して #ifdef などの条件と一緒に使用するシンボルを定義します。
-display Display	使用する X サーバーを指定します。 -screen オプション用に使用する画面も指定し、 -global オプション用にプリプロセッサ・シンボルが取り出される画面を指定します。
-edit FileName	指定した属性の内容を編集し、そこにリストされた値を置き換えて、所定のファイルに入れなければならないことを示します。これによって、デフォルトに対して行った変更をリソース・ファイルに納め、コメントやプリプロセッサ行を保管することができます。
-global	このオプションは、操作が画面独立 RESOURCE_MANAGER 属性上でのみ実行されなければならないことを示します。このオプションは X11R5 に固有なものです。
-help	使用可能なフラグの簡潔な説明を出力します。
-I Directory	(大文字の i) プリプロセッサへ渡して #include で参照されるファイルを検索するディレクトリを指定します。
-load	指定した属性の古い内容を置き換えて、新しい値として入力がロードされることを示します。これはデフォルトのアクションです。
-merge	指定した属性の現在の内容が置き換えられる代わりに、入力と組み合わせることを示します。このオプションを指定すると、2 つの入力は字句通りにソートされて組み合わせるので、ユーザーの希望通りではないかもしれませんが、下位互換性を保つために残されています。
-n	このオプションは、(-load または -merge とともに使用されている場合には) 指定された属性に対して、または (-edit とともに使用されている場合には) リソース・ファイルに対してなされた変更が、標準出力上に表示されても実行されないことを示します。このオプションは X11R5 に固有なものです。
-nocpp	xrdb コマンドで、入力ファイルを属性にロードする前に、そのファイルをプリプロセッサを通じて実行されないことを示します。
-override	指定した属性の現在の内容が置き換えられる代わりに、入力に追加されることを示します。新規の入力によって前の入力が上書きされます。
-query	指定した属性の現在の内容を標準出力に出力することを示します。入力リソース・ファイルの中のプリプロセッサ・コマンドは、入力ファイルの一部であり、属性の一部ではないので、このフラグからの出力には現れないことに注意してください。
-quiet	重複した入力についての警告が表示されないことを示します。このオプションは X11R5 に固有なものです。
-remove	指定した属性をサーバーから除去することを示します。
-retain	xrdb コマンドが最初のクライアントであれば、サーバーに再設定しないように指図することを示します。 xdm および xinit コマンドは、常に最初のクライアントとして動作するので、このフラグが通常の状態が必要になることは、決してありません。このオプションは X11R5 に固有なものです。
-screen	操作がディスプレイのデフォルト画面の SCREEN_RESOURCES 属性上でのみ実行されなければならないことを示します。このオプションは X11R5 に固有なものです。

項目	説明
-screens	操作がディスプレイの各画面の SCREEN_RESOURCES 属性上で実行されることを示します。 -load および -merge の場合は、入力ファイルは、各画面につき一度ずつ処理されます。このオプションは X11R5 に固有なものです。
-symbols	プリプロセッサ用に定義されたシンボルが、標準出力へ出力されることを示します。
-UName	このシンボルのすべての定義をプリプロセッサへ渡して除去します。

例

1. ファイルをデータベースにロードするには、下記のように入力します。

```
xrdb -load myfile
```

2. ロードしたばかりのデータベースの内容を取り出して、編集するか、または新しいファイルに入れるには、下記のように入力します。

```
xrdb -edit newfile
```

ファイル

xrdb コマンドは `~/Xdefaults` ファイルを汎用化します。

xsend コマンド

目的

セキュア通信チャンネル内で秘密メールを送信します。

構文

xsend *User*

説明

xsend コマンドは、指定した受信者しか読むことができないメッセージを送信します。このコマンドは **mail** コマンドと似ていますが、このコマンドで送信されたメールは、秘密メールとなります。

xsend コマンドは、秘密メールを受信するために **enroll** コマンドおよび **xget** コマンドと一緒に使われます。**enroll** コマンドを使用してパスワードを設定し、秘密メールを受信します。**xget** コマンドではメールを受信するためにそのパスワードを使用します。

xsend コマンドは、EOF (Ctrl-D) またはピリオド (.) が入力されるまでは、標準入力を読み取ります。そして、ヘッダー情報と一緒にそのテキストを暗号化して送信します。暗号化されたメッセージを送信した後、**xsend** コマンドは標準メール・メッセージを受信者へ返し、秘密メールを受信したことを知らせます。

注: 秘密メールはローカル・ユーザーにしか送信することができません。

例

1. 秘密メールを送信するには、次のように入力します。

```
xsend ron
```

受信者の名前を付けて **xsend** コマンドを発行すると、メッセージのテキストを入力するためにメール・システムが使われます。ユーザー `ron` へのメッセージを入力し終わったら、**Enter** キーの次に

Ctrl-D または (ピリオド) を押してメール・エディターを終了し、メッセージを送信します。**xsend** コマンドは、メッセージを送信する前に暗号化を行います。

2. ファイルを別のユーザーに送信するには、次のように入力します。

```
xsend lance <proposal
```

この例では、ファイル `proposal` がユーザー `lance` に送られます。

ファイル

項目	説明
<code>/var/spool/secretmail/*.keys</code>	User 用の暗号化された鍵が入っています。
<code>/var/spool/secretmail/*.{0-9}</code>	User 用の暗号化されたメール・メッセージが入っています。
<code>/usr/bin/xsend</code>	コマンド実行可能ファイルが入っています。

関連資料:

187 ページの『`xget` コマンド』

関連情報:

`bellmail` コマンド

`enroll` コマンド

`mail` コマンド

メール・アプリケーション

機密メールの送受信

xset コマンド

目的

X-Window 環境のオプションを設定します。

構文

```
xset [ -display Display ] [ b [ Volume [ Pitch [ Duration ] ] ] | -b | b on | b off ] [ bc | -bc ] c [ Volume ] | -c | c on | c off ] [ [ - | + ] fp [ - | + | = ] Path [ ,Path, [ ... ] ] ] [ fp default ] [ fp rehash ] [ [ - ] led [ Integer ] ] [ led on | led off ] [ m [ Accelerator ] [ Threshold ] ] ] [ m [ ouse ] default ] [ p Pixel Color ] [ [ - ] r ] [ r on | r off ] [ s [ Length [ Period ] ] ] [ s blank | s noblank ] [ s expose | s noexpose ] [ s on | s off ] [ s activate ] [ s reset ] [ s default ] [ q ]
```

説明

xset コマンドはユーザーの X-Window 環境をカスタマイズします。

フラグ

項目

-display *Host:Display*

b または **b on**

b [*Volume* [*Pitch* [*Duration*]]]

-b または **b off**

bc または **-bc**

c または **c on**

c *Volume*

-c または **c off**

fp=Path...

fp- または **-fp**

fp+ または **+fp**

fp default

説明

使用する X サーバーを指定します。サーバーの詳細については、**X** コマンドのセクションを参照してください。

ベル機能をオンにします。これはデフォルト設定です。

注: どのハードウェアでもベルの特性を変更できるとは限りませんが、変更できる場合は、すべての **b** フラグの変更とその変数を使用できます。

ベルの音量、ピッチ、および音長を指定します。このフラグには、下記の 3 つまでの数値を使用できます。

Volume 数値を 1 つしか指定しなければ、**Volume** であると想定されます。ベルの音量は、現在のハードウェアの機能に応じて、ベルの最大音量に対するパーセントを表す数値として設定してください。

Pitch 第 2 の数値はヘルツ単位で、ベルのトーンです。

Duration

第 3 の数値はミリ秒単位で、ベルを鳴らす時間の長さです。

ベル機能をオフにします。

使用可能であれば、サーバー内のバグ互換モードをコントロールします。先行 - (ハイフン) を付けると、このモードが使用不可になり、付けなければバグ互換モードが使用可能になります。**bc** フラグを機能させるには、サーバーで

MIT-SUNDRY-NONSTANDARD プロトコル拡張機能がサポートされなければなりません。

新しいアプリケーション開発は、バグ互換モードを使用不可にして実行する必要があります。

bc フラグは、X11 リリース 4 (X11R4) より前のクライアント向けに用意されています。X11R4 より前のクライアントによっては、各種プロトコル要求で無効な値を渡すものがあります。この種のクライアントは、X11R4 サーバーを使用して実行すると、異常終了するか、または正常に操作できません。

このフラグは、この種のクライアントも引き続き実行できるように、一定のバグを X サーバーに明示的に再び取り込みます。

クリック機能をオンにします。システム・デフォルト。

現在のハードウェア機能に応じて、クリックの最大許容値のパーセントを 0 から 100 までの数値で指定します。

クリック機能をオフにします。

Path パラメーターで与えられたディレクトリーにフォント・パスを設定します。ディレクトリーは、クライアントではなくサーバーによって解釈され、サーバーに依存します。サーバーは、**mkfontdir** コマンドで作成されたフォント・データベースが入っていないディレクトリーを無視します。**fp** フラグでサポートされるオプションと変数を、すべて使用することができます。

Path パラメーターで指定したフォント・パスを、**fp** の前に - (ハイフン) が付いている場合は現フォント・パスの最後から、**fp** の後に - (ハイフン) が付いている場合はそのフォント・パスの最初から削除します。

Path パラメーターで指定したフォント・パスを、**fp** の前に - (ハイフン) が付いている場合はフォント・リストの最後に追加し、**fp** の後に - (ハイフン) が付いている場合はフォント・パスの終わりに追加します。

フォント・パスをサーバーのデフォルトにリセットします。

項目	説明
s reset	スクリーンセーバーがアクティブになっていれば、それを活動停止にします。
s blank	バックグラウンド・パターンを表示せずに、ビデオをブランクにするように (ハードウェアで可能な場合) 設定変更を変更します。
s noblank	表示をブランクにせずに、パターンを表示するように設定変更を設定します。
s expose	ウィンドウ・エクスポージャー (サーバーがウィンドウの内容を破棄) ができるように、設定変更を設定します。
s noexpose	エクスポージャー・イベントを発生させずにサーバーが画面を再生成できない限り、スクリーンセーバーを使用不可にするように設定変更を設定します。
q	現在の設定に関する情報を報告します。

上記の設定は、ログアウト時にデフォルト値にリセットされません。

注: どの X-Window 環境でも、これらのオプションをすべて使用できるわけではありません。

例

- ベルの音量を中に設定し、トーンを 50 ヘルツに設定し、ベルが鳴る時間の長さを 50 ミリ秒に設定するには、下記のように入力します。

```
xset b 50,50,50
```

- /usr/lib/X11/fonts** ディレクトリーへのフォント・パスを設定するには、下記のように入力します。

```
xset fp= /usr/lib/x11/fonts
```

- サーバーに現行フォント・パス内のフォント・データベースを再度読み取らせるには、下記のように入力します。

```
xset fp rehash
```

- 現在の設定に関する情報を表示するには、下記のように入力します。

```
xset q
```

このコマンドにより、下記のような出力が生成されます。

```
Keyboard Control:
  auto repeat: on   key click percent: 0   LED mask: 00000000
  auto repeating keys: 0000000000000000
                        0000000000000000
                        0000000000000000
                        0000000000000000
  bell percent: 50   bell pitch: 400   bell duration: 100
```

```
Pointer Control:
  acceleration: 2 = 2 / 1   threshold: 4
```

```
Screen Saver:
  prefer blanking: no   allow exposures: no
  timeout: 0   cycle: 0
```

```
Colors:
  default colormap: 0x8006e   BlackPixel: 0   WhitePixel: 1
```

```
Font Path:
  /usr/lib/X11/fonts/,/usr/lib/X11/fonts/75dpi/,/usr/lib/X11/fonts/100dpi/,/usr/
  lib/X11/fonts/oldx10/,/usr/lib/X11/fonts/oldx11/,/usr/lib/X11/fonts/bmug/,/usr/l
  ib/X11/fonts/info-mac/,/usr/lib/X11/fonts/JP/,/usr/lib/X11/fonts/misc/
```

xsetroot コマンド

目的

X コマンドのルート・ウィンドウ・パラメーターを設定します。

構文

```
xsetroot [ -bg Color ] [ -cursor CursorFile MaskFile ] [ -cursor_name CursorName ] [ -def ] [ -display Display ] [ -fg Color ] [ -help ] [ -name String ] [ -rv ] [ -bitmap FileName | -gray | -grey | -mod X Y | -solid Color ]
```

説明

xsetroot コマンドを用いると、X が実行されているワークステーション・ディスプレイ上のバックグラウンド (ルート) ウィンドウの外観を、適宜変更することができます。通常、**xsetroot** コマンドで、希望する個別設定された外観になるまで吟味し、それから作成し終った **xsetroot** コマンドを X 始動ファイルの中へ保管します。オプションが指定されていないか、または **-def** フラグが指定されていると、ウィンドウはそのデフォルト状態にリセットされます。**-def** フラグは、他のフラグと一緒に使用することができ、デフォルト状態にリセットされるのは、他の特性が指定されていないときのみです。

バックグラウンド・カラー (タイル) 変更フラグ (**-bitmap**, **-solid**, **-gray**, **-grey**, **-mod**) を一度に 1 つだけ指定できます。

フラグ

項目

-bg *Color*

-bitmap *FileName*

-cursor *CursorFile MaskFile*

-cursor_name *CursorName*

-def

-display *Display*

-fg *Color*

-gray

-grey

-help

-mod *X Y*

説明

バックグラウンド・カラーの指定に *Color* パラメーターを使用します。ファイルに指定されたビットマップを用いてウィンドウ・パターンを設定します。ユーザーは、**bitmap** プログラムを使用して、独自のビットマップ・ファイル (小さいピクチャー) を作成できます。バックグラウンド全体は、ビットマップの反復タイルでできています。

ウィンドウの外部にあるときに、希望するポインター・カーソルに変更します。カーソルとマスク・ファイルは、**bitmap** プログラムを用いて作成可能なビットマップ (小さいピクチャー) です。マスクの動作に慣れるまでは、マスク・ファイルは、すべて黒が良いでしょう。ポインター・カーソルをカーソル・フォントからの標準カーソルのうちの 1 つに変更します。

指定されていない属性をデフォルト値にリセットします (再び見なれているグレイ・メッシュのバックグラウンドと、くぼんだ x 型のカーソルにします)。

接続するサーバーを指定します。X コマンドのセクションを参照してください。

Color パラメーターをフォアグラウンドとして使用します。フォアグラウンドおよびバックグラウンド・カラーは、**-cursor**, **-bitmap**, **-mod** フラグの場合のみ意味を持ちます。

バックグラウンド全体をグレイにします。

バックグラウンド全体をグレイにします。

使用方法メッセージを出力し、終了します。

ユーザーの画面上で、格子状のグリッド・パターンにします。X および Y パラメーターは、1 から 16 までの整数です。0 と負の数は、1 としで扱われます。

項目	説明
-name <i>String</i>	ルート・ウィンドウ名を <i>String</i> パラメーターに設定します。デフォルト値はありません。通常何らかの名前がウィンドウに割り当てられ、ウィンドウ・マネージャーは、ウィンドウがアイコン化されたときにテキスト表現を使用できるようになっています。ユーザーはバックグラウンドをアイコン化することができないので、このフラグは使用されません。
-rv	フォアグラウンド・カラーとバックグラウンド・カラーを交換します。通常は、フォアグラウンドは黒で、バックグラウンド・カラーは白です。
-solid <i>Color</i>	ルート・ウィンドウのバックグラウンドを指定された色に設定します。このフラグは、カラー・サーバー上でのみ使用されます。

XSS コマンド

目的

無人のワークステーションのセキュリティーを向上させます。

構文

```
xss [ -e CommandString ] [ -timeout Seconds ] [ -display DisplayPtr ] [ -v ] [ -fg Color ] [ -bg Color ] [ -geometry wxh+x+y ]
```

説明

xss コマンドは、新しく追加されたマサチューセッツ工科大学 (MIT) のスクリーンセーバー拡張機能とともに動作して、ユーザーがコントロール可能なスクリーンセーバーまたはスクリーン・ロックをインプリメントします。このコマンドは、無人のワークステーションのセキュリティーを改善するように設計されています。

xss コマンドは、スクリーンセーバー・タイムアウト・メッセージを受信したとき、またはユーザーがプッシュボタンを始動したときに、ユーザーが指定したコマンド文字列を実行します。ユーザーがコマンドを指定していなければ、**xss** コマンドはデフォルトで **xlock** コマンドを実行します。

注: **xss** コマンドは、新しく追加された MIT スクリーンセーバー拡張機能だけを使用します。**xss** コマンドは、古い X サーバー上では動作しません。また、古い X 拡張ライブラリーを使っているときも動作しません。

フラグ

項目	説明
-e <i>CommandString</i>	スクリーンセーバーがタイムアウトになったとき、またはユーザーがプッシュボタンを始動したときに、 xss コマンドを実行するように設定します。 <i>CommandString</i> パラメーター値が複数のワードの場合、その値を " " (二重引用符) で囲まなければならないことに注意してください。
-timeout <i>Seconds</i>	ユーザーのアクティビティーがないままスクリーンセーバーがタイムアウトになるまでの時間 (秒) を設定します。この時間が経過すると、 xss コマンドは <i>CommandString</i> パラメーターを実行します。
-display <i>DisplayPtr</i>	X11 ディスプレイへの接続を設定します。
-v	詳細モードをオンにします。
-fg <i>Color</i>	プッシュボタンのフォアグラウンド・カラーを設定します。
-bg <i>Color</i>	プッシュボタンのバックグラウンド・カラーを設定します。
-geometry <i>wxh+x+y</i>	クライアント・ウィンドウのサイズと位置を指定します。

例

xss コマンドに **-display** フラグを付けてリモートで実行している場合、**xss** コマンドで実行されるコマンドにも **-display** フラグ・オプションを指定しなければならないことに注意してください。下記のリモートでの実行の例を参照してください。

1. リモートでの実行

```
xss -display myhost:0 -e "xlock -remote -display myhost:0"
```

2. スクリーンセーバーのみ

```
xss -e "xlock -nolock"
```

3. 単純な例

```
xss -e xlock
```

xstr コマンド

目的

C プログラムから文字列を抽出して、共用文字列をインプリメントします。

構文

```
xstr [ -v ] [ -c ] [ - ] [ File ]
```

説明

xstr コマンドは、大きなプログラムのコンポーネント部分の中の文字列がハッシュされているファイル **strings** を保守します。これらの文字列はこの配列への参照に置き換えられます。これは、共用定数文字列を作成する上で役立ち、特に読み取り専用の文字列には有効です。

下記のコマンドは、

```
xstr -c File
```

File パラメーター内の C ソースから文字列を抽出し、フォーマット (**&xstr[number]**) の数値式を使って文字列参照を置き換えます。ファイルの前に **xstr** 配列の適切な宣言が置かれます。作成される C テキストはファイル **x.c** 内に入れられ、コンパイルされます。このファイルからの文字列は、まだ存在していなければ、ファイル **strings** に追加されます。文字列が繰り返し文字列および既存の文字列の接尾部である場合には、ファイル **strings** は変更されません。

文字列がファイル内の他の文字列の接尾部であっても、それより短い文字列が **xstr** コマンドによって最初に見つけられた場合は、その両方の文字列がファイル **strings** 内に入れられます。

大きなプログラムのすべてのコンポーネントがコンパイルされると、共通の **xstr** 配列スペースを宣言するファイル **xs.c** を、下記のフォーマットのコマンドを使って作成できます。

```
xstr
```

次に、この **xs.c** ファイルはコンパイルされ、プログラムの残りと一緒にロードされなければなりません。もし可能ならば、配列を読み取り専用 (共用) にして、スペースとスワップのオーバーヘッドを節約することができます。

xstr コマンドは単一ファイルに対して使うこともできます。下記のコマンドは、

```
xstr File
```

前のコマンドと同様に **x.c** と **xs.c** の 2 つのファイルを作成しますが、同じディレクトリー内の他の **strings** ファイルを使用したりそれに影響を与えたりしません。

マクロ定義によって文字列が作られたり、実際には必要ない文字列を含む条件コードがある場合には、C プリプロセッサの後から **xstr** コマンドを実行するのが有効になります。

xstr コマンドは、引数 **-** (ハイフン) が指定されているときは標準入力から読み取り、同時に **-c** フラグが指定されない限り、**strings** ファイルを変更することはありません。

C プリプロセッサの後から **xstr** コマンドを実行するのに適切なコマンド・シーケンスは、下記のとおりです。

```
cc -E name.c | xstr -c -
cc -c x.c
mv x.o name.o
```

新しい項目が追加されなければ、**xstr** コマンドはファイル **strings** にはタッチしません。そのため、本当に必要な場合でなければ、**make** コマンドによって **xs.o** ファイルを再作成することは避けられます。

フラグ

項目	説明
-c	指定されたファイルから文字列を取り出し、それを strings ファイル内に入れます。
-v	詳細モード。 strings ファイル内で文字列が見つかったとき、あるいは新しい文字列かどうかを知らせます。
-	標準入力から読み取ります。

例

1. *File.c* パラメーター内の C ソースから文字列を抽出し、フォーマット (**&xstr[number]**) の数値式を使って文字列参照を置き換えます。

```
xstr -c File.c
```

ファイルの前に **xstr** 配列の適切な宣言が置かれます。作成される C テキストはファイル **x.c** 内に入れられ、コンパイルされます。

2. **xs.c** ファイル内の共通 **xstr** 配列スペースを宣言するには、下記のように入力します。

```
xstr
```

ファイル

項目	説明
strings	抽出された文字列を含むファイル。
x.c	抽出された C ソース。
xs.c	xstr 配列の定義用 C ソース。
/tmp/xs*	xstr コマンドが strings ファイルにタッチしない場合の一時ファイル。
/usr/ccs/bin/mkstr	実行可能ファイル。
/usr/ccs/bin/mkstr	Berkeley 環境のための実行可能ファイル。

関連情報:

mkstr コマンド

xterm コマンド

目的

X Window System の端末エミュレーターを提供します。

注: **xterm** コマンドは、マサチューセッツ工科大学 (MIT) の X Window System のバージョン 11、リリース 6 から機能拡張なしに移植されたものです。 **xterm** コマンドは、グローバル化をサポートしていません。地域化または国際化された端末エミュレーターが必要な場合は、**aixterm** コマンドまたは **dtterm** コマンドを使用してください。

構文

```
xterm [ --Xtoolkitoption... ] [ -Option ... ]
```

説明

xterm プログラムは、X Window System の端末エミュレーターです。このウィンドウ・システムを直接使用できないプログラムのために DEC VT102 および Tektronix 4014 互換端末を提供します。基本となるオペレーティング・システムが端末サイズ変更機能をサポートしている場合、**xterm** プログラムはウィンドウのサイズが変更されると、その機能を使って、ウィンドウで実行中のプログラムにウィンドウが変更されたことを通知します。

VT102 および Tektronix 4014 端末はそれぞれ自分自身のウィンドウを持っているので、ユーザーは、1 つのウィンドウでテキストを編集しながら、同時に他のウィンドウでグラフィックスを見ることができません。正しい縦横比 (高さ/幅) を維持するために、Tektronix のグラフィックスは、ウィンドウに適合する 4014 縦横比において最大のボックスに制限されます。このボックスは、ウィンドウの左上に位置します。

両方のウィンドウを同時に表示できますが、キーボード入力や端末出力を受信するアクティブ・ウィンドウ と見なされるのは、そのうちの 1 つのみです。テキスト・カーソルが現れるのはこのウィンドウの中です。アクティブ・ウィンドウはエスケープ・シーケンスまたはメニューで選択することができます。このメニューは、VT102 ウィンドウの場合は「VT Options (VT オプション)」メニュー、4014 ウィンドウの場合は「Tek Options (Tek オプション)」メニューです。

エミュレーション

VT102 エミュレーションはほぼ完全ですが、低速スクロール、VT52 モード、明滅文字属性、または横倍角および縦倍角文字セットはサポートしていません。 **xterm** コマンドを使用して機能する **termcap** ファイル・エントリーには、**xterm**、**vt102**、**vt100**、および「ansi」があります。 **xterm** コマンドは自動的に **termcap** ファイル内でこのエントリーを順番に検索し、その後に **TERM** および **TERMCAP** 環境変数を設定します。

xterm の特殊機能の多くは、標準 VT102 エスケープ・シーケンスとは異なる 1 組のエスケープ・シーケンスを使って、プログラム制御下で修正できる可能性があります。

Tektronix 4014 エミュレーションもほぼ完全です。ウィンドウ・サイズに合わせた 12 ビット・グラフィックス・アドレッシングをサポートしています。さらに、4 つのフォント・サイズと 5 つのライン・タイプもサポートします。ライト・スルー (write-thru) モードとフォーカス解除 (defocused) モードはサポートしていません。

Tektronix のテキスト・コマンドおよびグラフィックス・コマンドは **xterm** コマンドによって内部的に記録されます。そして、COPY エスケープ・シーケンスを送信することで (または、後のセクションで説明

するように Tektronix メニューから)、記録されたコマンドをファイルに書き込むこともできます。このファイルの名前は、**COPYyy-MM-dd.hh:mm:ss** のフォーマットです。yy、MM、dd、hh、mm、および ss は、それぞれ、コピーが実行された年、月、日、時、分、および秒を表します (このファイルは **xterm** コマンドが始動されたディレクトリーに作成されます。ログイン **xterm** の場合はホーム・ディレクトリーに作成されます)。

その他の機能

xterm コマンドは、ポインターがウィンドウの中に入ったとき (つまり、そのウィンドウを選択したとき)、自動的にテキスト・カーソルを強調表示します。そして、ポインターがウィンドウから出たとき (つまり、そのウィンドウの選択を解除したとき)、強調表示を解除します。ウィンドウがフォーカス・ウィンドウの場合は、ポインターがどこにあっても、テキスト・カーソルは強調表示されます。

VT102 モードでは、ウィンドウのディスプレイ領域と同じサイズの代替画面バッファをアクティブ化または非アクティブ化するエスケープ・シーケンスがあります。代替画面バッファを始動すると、現在の画面は保管され、代替画面と置き換わります。ウィンドウの上部がスクロールされた行の保管は、通常の画面が復元されるまではできません。

termcap ファイルに **xterm** コマンドのエントリーを指定しておけば、**vi** コマンド・エディターで代替画面に切り替えて編集でき、**vi** を終了すると画面が復元されます。

VT102 モードおよび Tektronix モードには、ウィンドウの名前を変更するエスケープ・シーケンスがあります。

オプション

xterm 端末エミュレーターでは、標準 X ツールキットのすべてのコマンド・ライン・オプションと、さらに下記のオプションを使用できます (オプションが - でなく + で始まる場合、そのオプションはデフォルトの値に復元されます)。

項目	説明
-help	xterm コマンドのオプションを説明するメッセージを表示します。
-132	通常、80 桁モードと 132 桁モードを切り替える VT102 DECCOLM エスケープ・シーケンスは無視されます。このオプションを指定すると、DECCOLM エスケープ・シーケンスが認識され、 xterm ウィンドウのサイズが適切に変更されます。
-ah	xterm コマンドに、常にテキスト・カーソルを強調表示させます。デフォルトでは xterm コマンドは、フォーカスが失われたとき、またはポインターがウィンドウを離れたとき、テキスト・カーソルを強調表示しません。
+ah	xterm コマンドに、フォーカスに従ってテキスト・カーソルを強調表示させます。
-b Number	内部のボーダー (文字の外側の端とウィンドウのボーダーの距離) のサイズをピクセルで指定します。デフォルトは 2 です。指定された範囲が示すクラスを単語による選択用に設定します。
-cc CharacterClassRange:Value[...]	改行をライン・モード選択のときにカットしません。
-cn	改行をライン・モード選択のときにカットします。
+cn	テキスト・カーソルに使用する色を指定します。デフォルトでは、テキストと同じフォアグラウンド・カラーが使用されます。
-cr Color	xterm コマンドに、タブで始まるウィンドウと同じ幅の行を正しく表示できない (先頭のタブが表示されない) という more プログラムのバグを回避させます。もともとこのバグは curses 関数のカーソル移動パッケージによるものだと考えられていたので、このオプションの名前が付いています。
-cu	

項目	説明
+cu	xterm コマンドに、上記の more プログラムのバグを回避させません。
-e Program [Arguments]	xterm ウィンドウで実行するプログラム (およびそのコマンド・ライン引数) を指定します。 -T オプションおよび -n オプションをコマンド・ラインで指定していなければ、ウィンドウのタイトルとアイコンの名前もこのオプションに指定したプログラムの名前に設定します。 注: このオプションは、コマンド・ライン・オプションの最後に指定する必要があります。
-fb Font	太字テキストを表示するときに使用するフォントを指定します。このフォントは、通常のフォントと同じ高さと同幅を持つ必要があります。通常のフォントと太字フォントのいずれか 1 つしか指定していない場合、指定されたフォントを通常のフォントとして使用し、この場合の太字フォントは、指定されたフォントを重ね打ちすることで生成されます。デフォルトでは、通常のフォントが重ね打ちされます。
-i	useInsertMode リソースをオンにします。
+i	useInsertMode リソースをオフにします。
-j	xterm コマンドに高速スクロールを行わせません。通常、テキストは一度に 1 行ずつスクロールされます。このオプションを指定すると、 xterm コマンドは一度に複数行スクロールするため、表示が追いつかないという現象は起こりません。特に、膨大な量のテキストをスキャンする場合は、 xterm コマンドがきわめて速くなるため、このオプションを使用することを強くお勧めします。この機能は、低速スクロール機能を制御する VT100 エスケープ・シーケンスおよび「VT Options (VT オプション)」メニューからでも制御できます。
+j	xterm コマンドに高速スクロールを行わせません。
-ls	xterm ウィンドウで始動されるシェルがログイン・シェルであることを示します (つまり、 <i>ArgumentVector</i> パラメーターの最初の文字にハイフンを指定すると、そのシェルはユーザーの .login または .profile ファイルを読み取ります)。
+ls	始動されるシェルがログイン・シェルではないことを示します (つまり、通常のサブシェルになります)。
-mb	xterm コマンドにユーザーの入力が行の右端近くに達したときに、マージン・ベルを鳴らします。このオプションは、「VT Options (VT オプション)」メニューからでも制御できます。
+mb	マージン・ベルを鳴らしません。
-mc Milliseconds	マルチクリック選択間の最大時間間隔を指定します。
-ms Color	ポインター・カーソルに使用する色を指定します。デフォルトではフォアグラウンド・カラーが使用されます。
-nb Number	マージン・ベルを鳴らす (使用可能であれば) 行の右端からの文字数を指定します。デフォルトは 10 です。
-rw	反転ラップアラウンドを使用します。これによって、カーソルが行頭に来たときに、前行の行末に移動することができます。この機能は長いシェル・コマンド・ラインを編集するときに非常に便利なので、使用することをお勧めします。このオプションは、「VT Options (VT オプション)」メニューからでも制御できます。
+rw	反転ラップアラウンドを使用しません。

項目	説明
-aw	自動ラップア라운드を使用します。これにより、カーソルが行末に来てテキストが出力されたときに、次行の行頭に自動的に移動できます。
+aw	自動ラップア라운드を使用しません。
-s	xterm コマンドに非同期的にスクロールさせます。つまり、スクロール中に、画面は最新の情報を保持する必要がないことを意味します。このオプションを指定しておくと、ネットワーク待ち時間が長いときにも、 xterm コマンドは比較的高速に動作するので、通常は大規模なインターネットや多数のゲートウェイを介して実行するときに変有効です。
+s	xterm コマンドに同期的にスクロールさせます。
-sb	スクロールしてウィンドウの上に消えた行を保管し、消えた行を見るためのスクロール・バーを表示します。このオプションは、「VT Options (VT オプション)」メニューからでも制御できます。
+sb	スクロール・バーを表示しません。
-sf	ファンクション・キーに対して、Sun Function Key エスケープ・コードを生成します。
+sf	ファンクション・キーに対して、標準のエスケープ・コードを生成します。
-si	ウィンドウへ出力しても、画面の位置を自動的にスクロール領域の最下部まで移動させません。このオプションは、「VT Options (VT オプション)」メニューからでも制御できます。
+si	ウィンドウへ出力すると、ウィンドウは最下部までスクロールします。
-sk	テキストの前の行を見るためにスクロール・バーを使用中にキーを押すと、ウィンドウは自動的にスクロール領域の最下部の通常的位置に移動します。
+sk	スクロール・バーを使用中にキーを押しても、ウィンドウは移動しません。
-sl Number	スクロールされて画面の上から消えた行を保管する行数を指定します。デフォルトは 64 です。
-t	VT102 モードではなく Tektronix モードで xterm コマンドを始動します。「Options」メニューを使ってこの 2 つのウィンドウ・モードを切り替えることができます。
+t	VT102 モードで xterm コマンドを始動します。
-tm String	端末設定キーワードとそれに続く機能にバインドされた文字を指定します (stty プログラムに似ています)。指定できるキーワードは、 intr 、 quit 、 erase 、 kill 、 eof 、 eol 、 swtch 、 start 、 stop 、 brk 、 susp 、 dsusp 、 rprnt 、 flush 、 weras 、 lnext です。制御文字は ^Character (例えば、 ^c や ^u) で指定することができ、 ^? は Delete を示します。
-tn Name	TERM 環境変数に設定する端末タイプの名前を指定します。この端末タイプは termcap データベースに存在し、「 li# 」および「 co# 」エントリを持っている必要があります。
-ut	xterm コマンドに、 /etc/utmp システム・ログ・ファイルに記録を書き込ませません。
+ut	xterm コマンドに、 /etc/utmp システム・ログ・ファイルに記録を書き込ませます。
-vb	ベルを鳴らす代わりに、ビジュアル・ベルを使用します。Ctrl+G キー・シーケンス・シグナルを受信すると、端末のベルを鳴らす代わりに、ウィンドウをフラッシュさせます。
+vb	ビジュアル・ベルを使用しないことを指定します。
-wf	xterm コマンドに、サブプロセスを始動する前に、最初のウィンドウのマッピングが行われるまで待ち状態にさせます。そうすると、初期端末サイズ設定と環境変数が正確になります。この後の端末サイズの変更を受け取るのは、アプリケーションの責任です。

項目	説明
+wf	xterm コマンドに、サブプロセスを始動する前に待ち状態にさせません。
-C	ウィンドウにコンソール出力を受信させることを示します。このオプションは、すべてのシステムでサポートされているわけではありません。コンソール出力を受信するには、コンソール・デバイスのオーナーで、そのデバイスに対する読み取りおよび書き込み許可を持っている必要があります。コンソール画面の xdm 下で X-Window を実行している場合、このオプションを有効にするには、セッション始動プログラムおよびセッション・リセット・プログラムを使って、明示的にコンソール・デバイスの所有権を変更する必要がある場合もあります。
-Scn	スレーブ・モードで使用する疑似端末の名前の最後の 2 文字と、継承したファイル・ディスクリプターの番号を指定します。このオプションは「%c%c%d」と解析されます。このオプションによって、 xterm コマンドは既存プログラムの入出力チャネルとして使用できるようになりますが、専用のアプリケーションの中で使用される場合もあります。

下記のコマンド・ライン引数は、古いバージョンとの互換性を保つために提供されます。次のリリースでは X ツールキットが同じタスクを実行する標準オプションを提供するので、サポートされない可能性もあります。

項目	説明
%geom	Tektronix ウィンドウに適したサイズと位置を指定します。* tekGeometry リソースを指定する場合の省略形です。
#geom	アイコン・ウィンドウに適した位置を指定します。* iconGeometry リソースを指定する場合の省略形です。
-T String	xterm プログラムのウィンドウのタイトルを指定します。- title と同じです。
-n String	xterm プログラムのウィンドウのアイコン名を指定します。* iconName リソースを指定する場合の省略形です。このオプションは、ツールキット・オプションの -name (この後を参照) とは異なるので注意してください。デフォルトのアイコン名はアプリケーションの名前です。
-r	フォアグラウンド・カラーとバックグラウンド・カラーをスワップすることで、反転表示をシミュレートします。- rv と同じです。
-wNumber	ウィンドウを囲むボーダーの幅をピクセル単位で指定します。- borderwidth および -bw と同じです。

下記の標準 X ツールキット・コマンド・ライン引数は、通常 **xterm** コマンドで使用できます。

項目	説明
-bg Color	ウィンドウのバックグラウンドに使用する色を指定します。デフォルトは白です。
-bd Color	ウィンドウのボーダーに使用する色を指定します。デフォルトは黒です。
-bw Number	ウィンドウを囲むボーダーの幅をピクセル単位で指定します。
-fg Color	テキストの表示に使用する色を指定します。デフォルトは黒です。
-fn Font	通常のテキストの表示に使用するフォントを指定します。デフォルトは固定 (fixed) です。
-name Name	リソースを獲得するためのアプリケーション名を指定します。デフォルトの実行可能ファイルの名前ではありません。Name パラメーター中に .(ドット) または * (アスタリスク) 文字を指定しないでください。
-title String	ユーザーが選択すると、ウィンドウ・マネージャーによって表示されるウィンドウ・タイトルの文字列を指定します。デフォルトのタイトルは、コマンド・ラインに指定した -e オプション以降の文字列であり、それ以外の場合はアプリケーションの名前になります。
-rv	フォアグラウンド・カラーとバックグラウンド・カラーをスワップすることで、反転表示をシミュレートします。
-geometryGeometry	VT102 ウィンドウに適したサイズと位置を指定します。X コマンドのセクションを参照してください。
-display Display	接続する X サーバーを指定します。X コマンドのセクションを参照してください。

項目	説明
<code>-xrmResourceString</code>	使用するリソース文字列を指定します。個別のコマンド・ライン・オプションを持たないリソースの設定に特に有効です。
<code>-iconic</code>	xterm コマンドが、ウィンドウ・マネージャーにそれを通常のウィンドウとしてではなく、アイコンとして始動するように要求することを示します。

リソース

このプログラムではコア X ツールキットのすべてのリソース名とクラスに加え、下記のリソース名およびクラスも使用できます。

項目	説明
iconGeometry (IconGeometry クラス)	アプリケーションをアイコン化するときに適したサイズと位置を指定します。すべてのウィンドウ・マネージャーがこれに従う必要はありません。
termName (TermName クラス)	TERM 環境変数に設定する端末タイプ名を指定します。
title (Title クラス)	ウィンドウ・マネージャーがこのアプリケーションを表示するときに使用する文字列を指定します。
ttyModes (TtyModes クラス)	端末設定キーワードとそれにバインドされた文字列を指定します。指定できるキーワードは、 intr 、 quit 、 erase 、 kill 、 eof 、 eol 、 swtch 、 start 、 stop 、 brk 、 susp 、 dsusp 、 rprnt 、 flush 、 weras 、 lnext です。制御文字は ^Character (例えば、 ^c や ^u) で指定することができ、 ^? は Delete を示します。このリソースを使うと、 xterm ウィンドウを始動するたびに stty プログラムを実行させる必要がないので、デフォルトの端末設定を変更するときに特に便利です。
useInsertMode (useInsertMode クラス)	適切なエントリーを TERMCAP 環境変数に追加することにより、挿入モードの使用を強制します。これは、システム termcap が壊れている場合に便利です。デフォルトは false です。
utmpInhibit (UtmpInhibit クラス)	xterm がユーザーの端末を /etc/utmp に記録しようとするかどうかを指定します。
sunFunctionKeys (SunFunctionKeys クラス)	ファンクション・キーに対して、標準のエスケープ・シーケンスではなく、 Sun Function Key エスケープ・コードを生成するかどうかを指定します。
waitForMap (WaitForMap クラス)	サブプロセスを始動する前に、 xterm コマンドに初期ウィンドウのマッピングを待機させるかどうかを指定します。デフォルトは False です。

下記のリソースは、**vt100** ウィジェット (**VT100** クラス) の一部として指定します。

項目	説明
allowSendEvents (AllowSendEvents クラス)	キーとボタンの合成イベント (X プロトコル SendEvent 要求を使用して生成する) を解釈するか廃棄するかどうかを指定します。デフォルトは False (偽) で、合成イベントを廃棄することを意味します。このようなイベントは大きなセキュリティー・ホールを作成する可能性があるので注意してください。
alwaysHighlight (AlwaysHighlight クラス)	xterm コマンドに常にテキスト・カーソルを強調表示させるかどうかを指定します。デフォルトでは、ポインターがウィンドウの外に移動したとき、またはウィンドウが入力フォーカスを失ったとき、テキスト・カーソルは強調表示されません。
appcursorDefault (AppcursorDefault クラス)	True (真) の場合、カーソル・キーは最初にアプリケーション・モードになります。デフォルトは False です。
appkeypadDefault (AppkeypadDefault クラス)	True (真) の場合、キーボード・キーは最初にアプリケーション・モードになります。デフォルトは False です。
autoWrap (AutoWrap クラス)	自動循環を使用可能にするかどうかを指定します。デフォルトは True です。
bellSupressTime (BellSupressTime クラス)	ベル・コマンドの送信後、次のベルを抑制する時間をミリ秒単位で指定します。デフォルトは 200 です。ゼロ以外の値を指定すると、最初のベルの処理が完了したことをサーバーが報告するまで次のベルが抑制されます。この機能は、ビジュアル・ベルで特に便利です。
boldFont (BoldFont クラス)	重ね打ちの代わりに使用する太字フォントの名前を指定します。
c132 (C132 クラス)	VT102 DECCOLM エスケープ・シーケンスを使用するかどうかを指定します。デフォルトは False です。
charClass (CharClass クラス)	文字クラスのリストを指定します。このリストは、各要素がコマンドで区切られ、 [low-]high:value のフォーマットになっています。これは、カット・アンド・ペーストを行うときに、どの文字の組を同じ文字として扱うかを決定するのに使用します。 256 ページの『文字クラス』を参照してください。
curses (Curses クラス)	curses 関数の最後の桁のバグを回避するかどうかを指定します。デフォルトは False です。
cutNewline (cutNewline クラス)	false の場合、トリプル・クリックして行を選択しても、その行の最後の改行は含まれません。 true の場合、改行は選択されます。デフォルトは true です。
cutToBeginningofLines (CutToBeginningOfLine クラス)	false の場合、トリプル・クリックして行を選択すると、現行ワードの次からしか選択されません。 true の場合、行全体が選択されます。デフォルトは true です。
background (Background クラス)	ウィンドウのバックグラウンドに使用する色を指定します。デフォルトは白です。
foreground (Foreground クラス)	ウィンドウの中でテキスト表示に使用する色を指定します。通常テキストの色で表示されるすべてを色変更するには、インスタンス名よりクラス名を設定するほうが簡単です。デフォルトは黒です。
cursorColor (Foreground クラス)	テキスト・カーソルに使用する色を指定します。デフォルトは黒です。
eightBitInput (EightBitInput クラス)	True (真) の場合、キーボードからのメタキャラクター入力が 8 番目のビットをオンにした単一文字として表されます。 False (偽) の場合、メタキャラクターの前に ESC が付いた 2 文字シーケンスに変換されます。デフォルトは True です。
eightBitOutput (EightBitOutput クラス)	表示するときにホストから送信された 8 ビット文字をそのまま受け入れるか切り捨てるかを指定します。デフォルトは True です。
font (Font クラス)	通常のフォントの名前を指定します。デフォルトは固定 (fixed) です。
font1 (Font1 クラス)	1 番目の代替フォントの名前を指定します。
font2 (Font2 クラス)	2 番目の代替フォントの名前を指定します。
font3 (Font3 クラス)	3 番目の代替フォントの名前を指定します。
font4 (Font4 クラス)	4 番目の代替フォントの名前を指定します。

項目
font5 (Font5 クラス)
font6 (Font6 クラス)
geometry (Geometry クラス)
hpLowerleftBugCompat (**hpLowerleftBugCompat** クラス)

internalBorder (**BorderWidth** クラス)

jumpScroll (**JumpScroll** クラス)

項目

loginShell (**LoginShell** クラス)

marginBell (**MarginBell** クラス)

multiClickTime (**MultiClickTime** クラス)

multiScroll (**MultiScroll** クラス)

nMarginBell (**Column** クラス)

pointerColor (**Foreground** クラス)

pointerColorBackground (**Background** クラス)

pointerShape (**Cursor** クラス)

resizeGravity (**ResizeGravity** クラス)

reverseVideo (**ReverseVideo** クラス)

reverseWrap (**ReverseWrap** クラス)

saveLines (**SaveLines** クラス)

scrollBar (**ScrollBar** クラス)

scrollTtyOutput (**ScrollCond** クラス)

scrollKey (**ScrollCond** クラス)

scrollLines (**ScrollLines** クラス)

説明

5 番目の代替フォントの名前を指定します。

6 番目の代替フォントの名前を指定します。

VT102 ウィンドウに適したサイズと位置を指定します。

xdb のバグを修正するかどうかを指定します。これは、**termcap** を無視して常に **ESC F** を送信して左下隅に移動します。**true** にすると、**xterm** は、**ESC F** を画面の左下隅に移動する要求として解釈します。デフォルトは **false** です。

文字とウィンドウのボーダーの間の距離をピクセル単位で指定します。デフォルトは 2 です。

高速スクロールを使用するかどうかを指定します。デフォルトは **True** です。

説明

シェルが当該ウィンドウ内でログイン・シェルとして始動するかどうかを指定します。デフォルトは **False** です。

ユーザーが右マージン近くまで入力したときにベルを鳴らすかどうかを指定します。デフォルトは **False** です。

マルチクリック選択イベント間の最大時間をミリ秒単位で指定します。デフォルトは 250 ミリ秒です。

非同期でスクロールを行うかどうかを指定します。デフォルトは **False** です。

マージン・ベルを鳴らす (使用可能であれば) 右マージンからの文字数を指定します。

ポインタのフォアグラウンド・カラーを指定します。デフォルトは **XtDefaultForeground** です。

ポインタのバックグラウンド・カラーを指定します。デフォルトは **XtDefaultBackground** です。

ポインタの形状の名前を指定します。デフォルトは **xterm** です。

ウィンドウのサイズを (上下に) 変更したときの動作を指定します。**NorthWest** を指定すると、画面上の最上部のテキスト行が固定されます。ウィンドウを短くした場合、行は最下部から消えていきます。ウィンドウを長くした場合、ブランク行が最下部に追加されます。

これは、MIT バージョン X11R4 と互換性のある動作です。

SouthWest (デフォルト) を指定すると、画面の最下部のテキスト行が固定されます。ウィンドウを長くした場合、スクロールで画面から消えて保管されていた行がスクリーンに戻ってきます。ウィンドウを短くした場合、行は画面の最上部から消えていき、最上部の保管された行はドロップされます。

反転表示をシミュレートするかどうかを指定します。デフォルトは **False** です。

逆循環を使用するかどうかを指定します。デフォルトは **False** です。

スクロール・バーがオンになっているとき、画面の上から消えていく行の保管される行数を指定します。デフォルトは 64 です。スクロール・バーを表示するかどうかを指定します。デフォルトは **False** です。

端末へ出力すると、スクロール・バーが自動的にスクロール領域の最下部に移動するかどうかを指定します。デフォルトは **True** です。

キーを押すと、スクロール・バーが自動的にスクロール領域の最下部に移動するかどうかを指定します。デフォルトは **False** です。

scroll-back および **scroll-forw** アクションがデフォルトとして使用する行数を指定します。デフォルト値は 1 です。

項目	説明
signalInhibit (SignalInhibit クラス)	「Main Options (メイン・オプション)」メニュー内の xterm へシグナルを送信するエントリーを禁止するかどうかを指定します。デフォルトは False です。
tekGeometry (Geometry クラス)	Tektronix ウィンドウに適したサイズと位置を指定します。
tekInhibit (TekInhibit クラス)	Tektronix モードに入るエスケープ・シーケンスを無視するかどうかを指定します。デフォルトは False です。
tekSmall (TekSmall クラス)	形状が明示的に指定されていない場合に、Tektronix モード・ウィンドウを最小のサイズで始動するかどうかを指定します。これは画面の小さいディスプレイ上で xterm コマンドを実行するのに便利です。デフォルトは False です。
tekStartup (TekStartup クラス)	xterm を Tektronix モードで始動するかどうかを指定します。デフォルトは False です。
titeInhibit (TiteInhibit クラス)	xterm が TERMCAP 文字列から、 ti および te の termcap ファイル・エントリー (多数の画面指向プログラムの始動中に代替画面間を切り替えるのに使用するエントリー) を除去するかどうかを指定します。このリソースが設定されていると、 xterm コマンドは代替画面に切り替えるためのエスケープ・シーケンスも無視します。
translations (Translations クラス)	メニュー、選択、プログラムされた文字列などに対するキーとボタンの割り当てを指定します。詳しくは、アクションを参照してください。
visualBell (VisualBell クラス)	Ctrl+G キー・シーケンス・シグナルを受信したとき、ベル音を鳴らす代わりに、ビジュアル・ベル (明滅) を使用するかどうかを指定します。デフォルトは False です。

下記のリソースは、**tek4014** ウィジェット (**Tek4014** クラス) の一部として指定します。

項目	説明
width (Width クラス)	Tektronix ウィンドウの幅をピクセル単位で指定します。
height (Height クラス)	Tektronix ウィンドウの高さをピクセル単位で指定します。
fontLarge (Font クラス)	Tektronix ウィンドウで使用するラージ・フォントを指定します。
font2 (Font クラス)	Tektronix ウィンドウで使用するフォント番号 2 を指定します。
font3 (Font クラス)	Tektronix ウィンドウで使用するフォント番号 3 を指定します。
fontSmall (Font クラス)	Tektronix ウィンドウで使用するスモール・フォントを指定します。
initialFont (InitialFont クラス)	4 つの Tektronix フォントの中から最初に使用するフォントを指定します。値は set-tek-text アクションのものと同じです。デフォルトはラージ・フォントです。
ginTerminator (GinTerminator クラス)	GIN 報告または状況報告の後に続く文字を指定します。指定可能なのは次の 3 つです。「 none 」は終端文字を何も送信しません。「 CRonly 」は CR を送信します。「 CR&EOT 」は CR と EOT を送信します。デフォルトは none です。

さまざまなメニューに指定するリソースについては、**Athena SimpleMenu** ウィジェットに関する資料を参照してください。下記のリストは、各メニューのエントリーに対する名前とクラスです。

「**mainMenu**」には、下記のエントリーがあります。

項目

securekbd (SmeBSB クラス)
allowsends (SmeBSB クラス)
redraw (SmeBSB クラス)
line1 (SmeLine クラス)
suspend (SmeBSB クラス)

continue (SmeBSB クラス)

interrupt (SmeBSB クラス)
hangup (SmeBSB クラス)
terminate (SmeBSB クラス)
kill (SmeBSB クラス)
line2 (SmeLine クラス)
quit (SmeBSB クラス)

説明

secure() アクションを呼び出します。
allow-send-events(toggle) アクションを呼び出します。
redraw() アクションを呼び出します。
セパレーターです。
ジョブ制御をサポートするシステム上で **send-signal(tstp)** アクションを呼び出します。
ジョブ制御をサポートするシステム上で **send-signal(cont)** アクションを呼び出します。
send-signal(int) アクションを呼び出します。
send-signal(hup) アクションを呼び出します。
send-signal(term) アクションを呼び出します。
send-signal(kill) アクションを呼び出します。
セパレーターです。
quit() アクションを呼び出します。

「vtMenu」には、下記のエントリーがあります。

項目

scrollbar (SmeBSB クラス)
jumpscroll (SmeBSB クラス)
reversevideo (SmeBSB クラス)
autowrap (SmeBSB クラス)
reversewrap (SmeBSB クラス)
autolinefeed (SmeBSB クラス)
appcursor (SmeBSB クラス)
appkeypad (SmeBSB クラス)
scrollkey (SmeBSB クラス)
scrollttyoutput (SmeBSB クラス)
allow132 (SmeBSB クラス)
cursesemul (SmeBSB クラス)
visualbell (SmeBSB クラス)
marginbell (SmeBSB クラス)
altscreen (SmeBSB クラス)
line1 (SmeLine クラス)
softreset (SmeBSB クラス)
hardreset (SmeBSB クラス)
clearsavedlines (SmeBSB クラス)
line2 (SmeLine クラス)
tekshow (SmeBSB クラス)
tekmode (SmeBSB クラス)
vthide (SmeBSB クラス)

説明

set-scrollbar(toggle) アクションを呼び出します。
set-jumpscroll(toggle) アクションを呼び出します。
set-reverse-video(toggle) アクションを呼び出します。
set-autowrap(toggle) アクションを呼び出します。
set-reversewrap(toggle) アクションを呼び出します。
set-autolinefeed(toggle) アクションを呼び出します。
set-appcursor(toggle) アクションを呼び出します。
set-appkeypad(toggle) アクションを呼び出します。
set-scroll-on-key(toggle) アクションを呼び出します。
set-scroll-on-tty-output(toggle) アクションを呼び出します。
set-allow132(toggle) アクションを呼び出します。
set-cursesemul(toggle) アクションを呼び出します。
set-visualbell(toggle) アクションを呼び出します。
set-marginbell(toggle) アクションを呼び出します。
このエントリーは現在使用できません。
セパレーターです。
soft-reset() アクションを呼び出します。
hard-reset() アクションを呼び出します。
clear-saved-lines() アクションを呼び出します。
セパレーターです。
set-visibility(tek,toggle) アクションを呼び出します。
set-terminal-type(tek) アクションを呼び出します。
set-visibility(vt,off) アクションを呼び出します。

「fontMenu」には、下記のエントリーがあります。

項目

fontdefault (SmeBSB クラス)
font1 (SmeBSB クラス)
font2 (SmeBSB クラス)
font3 (SmeBSB クラス)
font4 (SmeBSB クラス)
font5 (SmeBSB クラス)
font6 (SmeBSB クラス)
fontescape (SmeBSB クラス)
fontsel (SmeBSB クラス)

説明

set-vt-font(d) アクションを呼び出します。
set-vt-font(1) アクションを呼び出します。
set-vt-font(2) アクションを呼び出します。
set-vt-font(3) アクションを呼び出します。
set-vt-font(4) アクションを呼び出します。
set-vt-font(5) アクションを呼び出します。
set-vt-font(6) アクションを呼び出します。
set-vt-font(e) アクションを呼び出します。
set-vt-font(s) アクションを呼び出します。

「tekMenu」には、下記のエントリーがあります。

項目	説明
tektextlarge (SmeBSB クラス)	set-tek-text(l) アクションを呼び出します。
tektext2 (SmeBSB クラス)	set-tek-text(2) アクションを呼び出します。
tektext3 (SmeBSB クラス)	set-tek-text(3) アクションを呼び出します。
tektextsmall (SmeBSB クラス)	set-tek-text(s) アクションを呼び出します。
line1 (SmeLine クラス)	セパレーターです。
tekpage (SmeBSB クラス)	tek-page() アクションを呼び出します。
tekreset (SmeBSB クラス)	tek-reset() アクションを呼び出します。
tekcopy (SmeBSB クラス)	tek-copy() アクションを呼び出します。
line2 (SmeLine クラス)	セパレーターです。
vtshow (SmeBSB クラス)	set-visibility(vt,toggle) アクションを呼び出します。
vtmode (SmeBSB クラス)	set-terminal-type(vt) アクションを呼び出します。
tekhide (SmeBSB クラス)	set-visibility(tek,toggle) アクションを呼び出します。

下記のリソースは、**Athena Scrollbar** ウィジェットを指定するときに使用します。

項目	説明
thickness (Thickness クラス)	スクロール・バーの幅をピクセル単位で指定します。
background (Background クラス)	スクロール・バーのバックグラウンドに使用する色を指定します。
foreground (Foreground クラス)	スクロール・バーのフォアグラウンドに使用する色を指定します。
	スクロール・バーのつまみ は、フォアグラウンド・カラーとバックグラウンド・カラーがピクセルごとに交互に並んだ単純なパターンです。

ポインターの使用方法

VT102 ウィンドウを作成した後、**xterm** コマンドを使用して、テキストを選択し、同一ウィンドウまたは異なるウィンドウにそれをコピーすることができます。

選択機能が呼び出されるのは、ポインター・ボタンを修飾子を付けずに使用したとき、および **Shift** キーを押しながらポインター・ボタンを使用したときです。キーおよびボタンへの機能の割り当ては、リソース・データベースで変更することができます。

ポインター・ボタン 1 (通常は左ボタン) は、テキストをカット・バッファーに保管するのに使用します。テキストの最初にカーソルを移動し、そこでボタンを押して、ボタンを押したまま、選択したい領域の最後までカーソルを移動し、そこでボタンを離します。選択されたテキストは強調表示され、グローバル・カット・バッファーに保管され、ボタンを離したときに **PRIMARY** 選択が作成されます。

ダブルクリックするとワードを、3 回クリックすると行を選択できます。4 回クリックすると元 (文字) に戻ります。マルチクリックは、ボタンを離してからボタンを押し下げるまでの時間によって決まります。したがって、選択の途中で選択の単位を変更することができます。キーまたはボタンの割り当てによって **X** 選択の作成が指定される場合、**xterm** コマンドは選択のオーナーである限り、選択されたテキストを強調表示したままにします。

ポインター・ボタン 2 (通常は中央のボタン) は、**PRIMARY** 選択にテキストがあればそれを、「タイプ」(ペースト) します。**PRIMARY** 選択にテキストがない場合、カット・バッファーからキーボード入力として挿入します。

ポインター・ボタン 3 (通常は右ボタン) は、現在の選択を拡張します。選択の左端より右端に近いところでこのボタンを押すと、選択の右側を拡張または縮小します。選択を右側から縮小していき、選択の左端に達すると、**xterm** コマンドは選択の左端を拡張するものと認識し、選択を元の状態に戻し、選択の左端を拡張または縮小します。

その逆も適用されます。選択の右端より左端に近いところでこのボタンを押すと、選択の左側を拡張または縮小します。選択の左側から縮小していき、選択の右端に達すると、**xterm** コマンドは選択の右端を拡張するものと認識し、選択を元の状態に戻し、選択の右端を拡張または縮小します。拡張は、最後の選択または拡張が行われたときの選択単位モードで開始されます。マルチクリックで選択単位を変更することができます。

後書きの改行がないテキストをカット・アンド・ペーストすることによって、異なるウィンドウの複数の場所からテキストを取り出し、例えば、シェルへのコマンドを形成したり、プログラムの出力を取り出して好きなエディターの中に挿入したりすることができます。カット・バッファは異なるアプリケーション間でグローバルに共用できるため、カット・バッファはその内容が分かっている「ファイル」と見なします。端末エミュレーターおよび他のテキスト・プログラムでは、カット・バッファをテキスト・ファイルとして扱います。つまりテキストは改行で区切られます。

スクロール領域には、現在ウィンドウに表示されている (強調表示されている) テキストについて、実際に保管されているテキストに対する相対的な位置と量が表示されます。より多く (最大まで) のテキストを保管するに従い、強調表示された領域のサイズは小さくなります。

スクロール領域にポインターを合わせてボタン 1 をクリックすると、隣接する行をディスプレイ・ウィンドウの最上部に移動します。

ボタン 3 をクリックすると、ディスプレイ・ウィンドウの最上部の行をポインターの位置まで下げます。

ボタン 2 をクリックすると、スクロール・バーのポインターの位置に対応する、保管されたテキストの位置にディスプレイを移動します。

VT102 ウィンドウと異なり、Tektronix ウィンドウではテキストのコピーはできません。Tektronix GIN モードでは可能ですが、このモードではカーソルが矢印から十字に変わります。どれかキーを押すと、そのキーと十字カーソルの現在の座標が送信されます。ボタン 1、2、3 をそれぞれクリックすると、おのおの文字 l、m、r が戻されます。テキストのコピーはできません。

Shift キーを押しながらポインター・ボタンを押すと、それぞれに対応する大文字が送信されます。ポインター・ボタンをキーと区別するには、文字の上位ビットを設定します (しかし、このビットは通常、端末モードが RAW でないと、取り除かれます。詳細は、**tty** コマンドのセクションを参照してください)。

メニュー

xterm コマンドには、4 つのメニューがあります。「mainMenu」、「vtMenu」、「fontMenu」、および「tekMenu」です。各メニューは、キーとボタンを正しく組み合わせて押すと開きます。ほとんどのメニューは、横線で 2 つの部分に分けられています。上半分には変更できる各種モードが入っています。現在アクティブなモードの隣にチェック・マークが表示されます。モードを 1 つ選択すると、その状態が切り替わります。メニューの下半分にはコマンド・エントリーが入っています。コマンド・エントリーを 1 つ選択すると、選択された機能が実行されます。

ウィンドウ内で Ctrl キーを押しながらポインター・ボタン 1 を押すと、「xterm」メニューが開きます。「mainMenu」には、VT102 および Tektronix ウィンドウの両方に適用される項目が入っています。**Secure Keyboard** モードは、パスワードを入力したり、セキュリティーが整備されていない環境で重要なデータを入力したりするときに使用されます。

メニューのコマンド・セクション内で注目すべきエントリーは、「Continue」、「Suspend」、「Interrupt」、「Hangup」、「Terminate」、「Kill」です。それぞれ「SIGCONT」、「SIGTSTP」、「SIGINT」、「SIGHUP」、「SIGTERM」、「SIGKILL」シグナルを、**xterm** (通常はシェル) 下で実

行中のプロセスのプロセス・グループに送信します。 **Continue** 機能は、ユーザーが誤って **Ctrl+Z** を押してプロセスを中断してしまったときに特に便利です。

「vtMenu」は、VT102 エミュレーションのさまざまなモードを設定します。このメニューは、VT102 ウィンドウ内で **Ctrl** キーを押しながらポインター・ボタン 2 を押すと開かれます。このメニューのコマンド・セクションにおいて、「soft reset」エントリーはスクロール領域をリセットします。この機能は、あるプログラムでスクロール領域が間違えて設定されたままになっている場合に、特に便利です (VMS や TOPS-20 を使っているときにしばしば発生する問題です)。

「full reset」エントリーは、画面を消去し、タブを 8 桁ごとにリセットし、端末モード (循環や低速スクロールなど) を、**xterm** コマンドがコマンド・ライン・オプションを処理し終えた直後の初期状態にリセットします。

「fontMenu」は、VT102 ウィンドウで使用するフォントを設定します。デフォルトのフォントとリソースで設定したいくつかの代替フォントに加えて、このメニューでは、**Set Font** エスケープ・シーケンス (制御シーケンスのセクションを参照) で最後に指定したフォントを、また (PRIMARY 選択が所有されている場合) 現在選択されているフォント名をフォント名として提供します。

「tekMenu」は、Tektronix エミュレーションのさまざまなモードを設定します。このメニューは、**Ctrl** キーを押しながらポインター・ボタン 2 を押すと開かれます。このメニューのモード・セクションでは、現在選択されているフォント・サイズが検査されています。コマンド・セクションの「**PAGE**」エントリーは、「Tektronix」ウィンドウを消去します。

セキュリティ

X-Window 環境には、さまざまな種類のセキュリティがあります。MIT サーバーは、**xdm** 下で実行され、多くの人に納得のいくセキュリティ・レベルを提供する「*magic cookie*」許可スキームを使用することができます。ユーザーのサーバーがサーバーへのアクセスを制御するためにホスト・ベースの機構しか使用していない場合 (**xhost** コマンドのセクションを参照)、さらにユーザーがアクセスするホスト上で他のユーザーがクライアントを実行できる場合、誰かが X プロトコルの基本サービスを使うアプリケーションを実行して、ユーザーの行動をのぞき見して、キーボードからの入力をすべてコピーする可能性があります。

これは、パスワードや他の重要なデータを入力するときに特に関係してきます。この問題に対する最善の解決方法は、ホスト・ベース制御より安全性の高い許可機構を使用することですが、**xterm** コマンドの中には、キーボード入力を保護するための簡単な機構があります。

「xterm」メニューには、「**Secure Keyboard**」エントリーがあります。このエントリーを使用可能にすると、すべてのキーボード入力が (**GrabKeyboard** プロトコル要求を使用して) **xterm** コマンドにのみ送信されます。アプリケーションがパスワード (またはその他の重要なデータ) の入力を要求するプロンプトを表示したときに、メニューを使って **Secure Keyboard** を使用可能にし、データを入力し、最後に、もう一度メニューを使って **Secure Keyboard** を使用不可にします。

一度にキーボードを保護できる X クライアントは 1 つだけなので、**Secure Keyboard** を使用可能にしようとしても失敗することもあります。このような場合、ベルが鳴ります。「**Secure Keyboard**」が成功すると、フォアグラウンド・カラーとバックグラウンド・カラーが入れ替わります (「Modes (モード)」メニューの「**Reverse Video**」エントリーを選択したときと同じようになります)。「**Secure Keyboard**」を終了すると、再び色は入れ替わります。色が切り替わらなかった場合は、スプーフィングされている可能性を疑ってください。

ユーザーが実行しているアプリケーションがパスワードの入力を求める前にプロンプトを表示する場合、そのプロンプトが表示される前に **Secure** モードにしておくのが一番安全です。つまり、そのプロンプトが正しく (つまり、反転された色で) 表示されているかどうかを確認するようにすれば、盗まれる可能性は最小限に抑えられます。また、もう一度メニューを出し、エントリーの隣にチェック・マークが表示されていることを確認することもできます。

Secure Keyboard モードで `xterm` ウィンドウをアイコン化した (またはマップ解除した) 場合、またはウィンドウ・マネージャーの再生 (ウィンドウのタイトル・バーまたは他の装飾部から選択する) を始動すると、**Secure Keyboard** モードは、自動的に使用不可になります (これが X プロトコルが簡単に壊れない理由の 1 つです)。このような状態が発生した場合、フォアグラウンド・カラーとバックグラウンド・カラーが元の状態に入れ替わり、警告のベルが鳴ります。

文字クラス

マウスの中央のボタンを 2 回連続で素早くクリックすると、同じクラスのすべての文字 (文字、ホワイト・スペース、句読点) が選択されます。人によって選択したいものが異なるため (例えば、ファイル名全体を選択するか、または個別のサブネームだけを選択するかなど)、デフォルトのマッピングは **charClass** (**CharClass** クラス) リソースを使って指定変更することができます。

このリソースは、**範囲: 値** という対がコンマで区切られて連なったものです。範囲は、設定する文字の ASCII コードに対応する 0 から 127 までの数字で、1 つの場合はその数字を、範囲を示す場合は小さい数字 (*low*)-大きい数字 (*high*) というフォーマットで指定します。値 は任意ですが、デフォルト・テーブルはリスト内で発生する最初の文字の文字番号を使います。

デフォルトのテーブルは下記のとおりです。

```
static int charClass[128] = {
/* NUL SOH STX ETX EOT ENQ ACK BEL */
 32,  1,  1,  1,  1,  1,  1,  1,
/* BS  HT  NL  VT  NP  CR  SO  SI */
  1, 32,  1,  1,  1,  1,  1,  1,
/* DLE DC1 DC2 DC3 DC4 NAK SYN ETB */
  1,  1,  1,  1,  1,  1,  1,  1,
/* CAN  EM  SUB  ESC  FS  GS  RS  US */
  1,  1,  1,  1,  1,  1,  1,  1,
/* SP  !  "  #  $  %  &  ' */
 32, 33, 34, 35, 36, 37, 38, 39,
/* (  )  *  +  ,  -  .  / */
 40, 41, 42, 43, 44, 45, 46, 47,
/* 0  1  2  3  4  5  6  7 */
 48, 48, 48, 48, 48, 48, 48, 48,
/* 8  9  :  ;  <  =  >  ? */
 48, 48, 58, 59, 60, 61, 62, 63,
/* @  A  B  C  D  E  F  G */
 64, 48, 48, 48, 48, 48, 48, 48,
/* H  I  J  K  L  M  N  O */
 48, 48, 48, 48, 48, 48, 48, 48,
/* P  Q  R  S  T  U  V  W */
 48, 48, 48, 48, 48, 48, 48, 48,
/* X  Y  Z  [  ¥  ]  ^  _ */
 48, 48, 48, 91, 92, 93, 94, 48,
/* `  a  b  c  d  e  f  g */
 96, 48, 48, 48, 48, 48, 48, 48,
/* h  i  j  k  l  m  n  o */
 48, 48, 48, 48, 48, 48, 48, 48,
/* p  q  r  s  t  u  v  w */
 48, 48, 48, 48, 48, 48, 48, 48,
/* x  y  z  {  |  }  ~ DEL */
 48, 48, 48, 123, 124, 125, 126, 1};
```

例えば、33:48,37:48,45-47:48,64:48 という文字列は、感嘆符、パーセント記号、ハイフン、ピリオド、スラッシュ、および & 文字が、文字や番号と同じように扱われることを表します。これは、電子メールのアドレスやファイル名をカット・アンド・ペーストするとき便利です。

アクション

vt100 または **tek4014** ウィジェットの変換を変更することにより、キー（またはキーのシーケンス）を任意の文字列の入力用に再度割り当てることができます。キーおよびボタン・イベント以外のイベント用に変換を変更することは望ましくなく、それをすると思わぬ動作を引き起こします。下記のアクションは、**vt100** または **tek4014** 変換リソース内での使用のために提供されます。

項目	説明
bell ([Percent])	基本量の指定したパーセンテージ分だけ上下した時点で、キーボード・ベルを鳴らします。
ignore ()	イベントを無視しますが、特殊ポインター位置のエスケープ・シーケンスを検査します。
insert ()	押されたキーに関連する文字または文字列を挿入します。
insert-seven-bit ()	insert () の同義語です。
insert-eight-bit ()	押されたキーに関連する文字または文字列の 8 ビット (メタ) バージョンを挿入します。実際のアクションは、 eightBitInput リソースの値によって異なります。
insert-selection (SourceName [, ...])	<i>SourceName</i> パラメーターが示す選択部分またはカット・バッファ内で検索した文字列を挿入します。選択元は、1 つ見つかるまで指定された順番で検査されます (大文字と小文字は区別されます)。通常使用される選択には、PRIMARY、SECONDARY、CLIPBOARD があります。カット・バッファは通常、CUT_BUFFER0 から CUT_BUFFER7 まであります。
keymap (Name)	リソース名が <i>Name</i> で、接尾部が <i>Keymap</i> (大文字と小文字は区別する) の新しい変換テーブルを動的に定義します。None という名前を指定すると、元の変換テーブルを復元します。
pop-up menu (MenuName)	指定したポップアップ・メニューを表示します。有効な名前 (大文字と小文字は区別されます) には、「mainMenu」、「vtMenu」、「fontMenu」、「tekMenu」があります。
secure ()	セキュリティーというセクションで説明されている Secure Keyboard モードを切り替えます。これらのモードは「mainMenu」の「securekbd」エントリから呼び出されます。
select-start ()	現在のポインターの位置からテキストの選択を開始します。選択方法については、ポインターの使用法のセクションを参照してください。
select-extend ()	ポインターをトラッキングし、選択部分を拡張します。これは Motion イベントにのみ有効です。
select-end (DestName [, ...])	現在選択しているテキストを、 <i>DestName</i> で指定したすべての選択部分またはカット・バッファへ入れます。
select-cursor-start ()	現在のテキスト・カーソル位置から選択を開始する点を除いて、 select-start に似ています。
select-cursor-end (DestName [, ...])	select-cursor-start と一緒に使用する必要があることを除いて、 select-end に似ています。

項目

set-vt-font(*d/1/2/3/4/5/6/e/s* [,*NormalFont* [, *BoldFont*]])

start-extend()

start-cursor-extend()

string(*String*)

scroll-back(*Count* [,*Units*])

scroll-forw(*Count* [,*Units*])

allow-send-events(*On/Off/Toggle*)

redraw()

send-signal(*SigName*)

説明

現在 VT102 ウィンドウで使用中のフォントを設定します。最初の引数は、使用するフォントを指定する単一文字です。

d または *D* は、デフォルトのフォント (**xterm** コマンド始動時に使用される初期フォント) を示します。

1 から 6 までの数字は、*font1* から *font6* までのリソースで指定したフォントを示します。

e または *E* は、エスケープ・コードを通じて設定された通常のフォントと太字フォントを示します (通常のフォントは 2 番目のアクション引数で、太字フォントは 3 番目のアクション引数でも指定することができます)。

s または *S* は、2 番目のアクション引数で指定されたフォント選択 (**xfonset** などのプログラムで作成された) を示します。

select-start に似ていますが、現在のポインター位置に選択部分が展開されます。

select-extend に似ていますが、現在のテキスト・カーソル位置に選択部分が展開されます。

入力されたかのように、指定したテキスト文字列を挿入します。文字列にホワイト・スペース、非英数字などの文字が入っている場合は、引用符で囲む必要があります。文字列引数が「0x」文字で始まる場合、その文字列は 16 進の文字定数と解釈されます。テキスト・ウィンドウを上方にスクロールし、以前スクロールして画面最上部から消えたテキストを表示します。*Count* 引数は、スクロールの単位 (*page*、*halfpage*、*pixel*、*line* のいずれか) を示します。

scroll-back に似ていますが、スクロールの方向が逆です。

allowSendEvents リソースを設定または切り替えます。これは、「mainMenu」の「allowsends」エントリから呼び出すこともできます。

ウィンドウを再描画します。これは、「mainMenu」の「redraw」エントリから呼び出すこともできます。

SigName で指定されたシグナルを **xterm** サブプロセス (**-e** コマンド・ライン・オプションで指定したシェルまたはプログラム) へ送信します。これは、「mainMenu」の「suspend」、「continue」、「interrupt」、「hangup」、「terminate」、「kill」エントリから呼び出すこともできます。使用できるシグナルの名前 (大/小文字を問わない) は、以下のとおりです。

「tstp」(オペレーティング・システムがサポートしている場合)、

「suspend」(tstp と同じ)、

「cont」(オペレーティング・システムがサポートしている場合)、

「int」、

「hup」、

「term」、

「quit」、

「alarm」、

「alarm」(「alarm」と同じ)、および

「kill」です。

項目
quit()

説明
SIGHUP をサブプログラムへ送信し、終了します。これは、「mainMenu」の「**quit**」エントリーから呼び出すこともできます。

項目

set-scrollbar(On/Off/Toggle)

説明

scrollbar リソースを切り替えます。これは、「vtMenu」の「**scrollbar**」エントリーから呼び出すこともできます。

set-jumpscroll(On/Off/Toggle)

jumpscroll リソースを切り替えます。これは、「vtMenu」の「**jumpscroll**」エントリーから呼び出すこともできます。

set-reverse-video(On/Off/Toggle)

reverseVideo リソースを切り替えます。これは、「vtMenu」の「**reversevideo**」エントリーから呼び出すこともできます。

set-autowrap(On/Off/Toggle)

長い行の自動折り返しを切り替えます。これは、「vtMenu」の「**autowrap**」エントリーから呼び出すこともできます。

set-reversewrap(On/Off/Toggle)

reverseWrap リソースを切り替えます。これは、「vtMenu」の「**reversewrap**」エントリーから呼び出すこともできます。

set-autolinefeed(On/Off/Toggle)

改行の自動挿入を切り替えます。これは、「vtMenu」の「**autolinefeed**」エントリーから呼び出すこともできます。

set-appcursor(On/Off/Toggle)

アプリケーション・カーソル・キー・モードの処理を切り替えます。これは、「vtMenu」の「**appcursor**」エントリーから呼び出すこともできます。

set-appkeypad(On/Off/Toggle)

アプリケーション・キーパッド・モードの処理を切り替えます。これは、「vtMenu」の「**appkeypad**」エントリーから呼び出すこともできます。

set-scroll-on-key(On/Off/Toggle)

scrollKey リソースを切り替えます。これは、「vtMenu」の「**scrollkey**」エントリーから呼び出すこともできます。

set-scroll-on-tty-output(On/Off/Toggle)

scrollTtyOutput リソースを切り替えます。これは、「vtMenu」の「**scrollttyoutput**」エントリーから呼び出すこともできます。

set-allow132(On/Off/Toggle)

c132 リソースを切り替えます。これは、「vtMenu」の「**allow132**」エントリーから呼び出すこともできます。

set-cursesemul(On/Off/Toggle)

curses リソースを切り替えます。これは、「vtMenu」の「**cursesemul**」エントリーから呼び出すこともできます。

set-visual-bell(On/Off/Toggle)

visualBell リソースを切り替えます。これは、「vtMenu」の「**visualbell**」エントリーから呼び出すこともできます。

set-marginbell(On/Off/Toggle)

marginBell リソースを切り替えます。これは、「vtMenu」の「**marginbel**」エントリーから呼び出すこともできます。

set-altscreen(On/Off/Toggle)

代替画面と現行画面を切り替えます。

soft-reset()

スクロール領域をリセットします。これは、「vtMenu」の「**softreset**」エントリーから呼び出すこともできます。

hard-reset()

スクロール領域、タブ、ウィンドウ・サイズ、カーソル・キーをリセットし、画面をクリアします。このアクションは、「vtMenu」の「**hardreset**」エントリーからでも呼び出されません。

clear-saved-lines()

hard-reset (上記のエントリーを参照) を実行し、スクリーンの上部から見えなくなった、保存された行の履歴をクリアします。これは、「vtMenu」の「**clearsavedlines**」エントリーから呼び出すこともできます。

set-terminal-type(Type)

Type 文字列に従って、出力先を vt ウィンドウ、または tek ウィンドウにします。このアクションは、「vtMenu」の「**tekmode**」エントリー、および「tekMenu」の「**vtmode**」エントリーから呼び出すこともできます。

set-visibility(vt/tek, On/Off/Toggle)

vt ウィンドウまたは tek ウィンドウを表示するかどうかを制御します。これは、「vtMenu」の「**tekshow**」および「**vthide**」エントリー、または「tekMenu」の「**vtshow**」および「**tekhide**」エントリーから呼び出すこともできます。

項目	説明
set-tek-text (<i>large/2/3/small</i>)	Tektronix ウィンドウで使用されるフォントを設定します。設定する値は、引数に従って、 tektextlarge 、 tektext2 、 tektext3 、 tektextsmall リソースの値です。これは、「tekMenu」のリソース名と同じエントリーから呼び出すこともできます。
tek-page ()	Tektronix ウィンドウをクリアします。これは、「tekMenu」の「 tekpage 」エントリーから呼び出すこともできます。
tek-reset ()	Tektronix ウィンドウをリセットします。これは、「tekMenu」の「 tekreset 」エントリーから呼び出すこともできます。
tek-copy ()	現行ウィンドウの内容を生成するために使用されるエスケープ・コードを、 COPY という名前で始まる現行ディレクトリー内のファイルへコピーします。これは、「tekMenu」の「 tekcopy 」エントリーから呼び出すこともできます。
visual-bell ()	ウィンドウを素早く点滅させます。

Tektronix ウィンドウには、下記のアクションもあります。

項目	説明
gin-press (<i>l/L/m/M/r/r</i>)	指定されたグラフィックス入力コードを送信します。

VT102 ウィンドウにおけるデフォルトの割り当ては、下記のとおりです。

Shift <KeyPress> Prior:	scroll-back(1,halpage) ¥n¥
Shift <KeyPress> Next:	scroll-forw(1,halpage) ¥n¥
Shift <KeyPress> Select:	select-cursor-start ¥ select-cursor-end(PRIMARY, CUT_BUFFER0) ¥n¥
Shift <KeyPress> Insert:	insert-selection(PRIMARY, CUT_BUFFER0) ¥n¥
~Meta<KeyPress>:	insert-seven-bit ¥n¥
Meta<KeyPress>:	insert-eight-bit ¥n¥
!Ctrl <Btn1Down>:	pop-up menu(mainMenu) ¥n¥
!Lock Ctrl <Btn1Down>:	pop-up menu(mainMenu) ¥n¥
~Meta <Btn1Down>:	select-start ¥n¥
~Meta <Btn1Motion>:	select-extend ¥n¥
!Ctrl <Btn2Down>:	pop-up menu(vtMenu) ¥n¥
!Lock Ctrl <Btn2Down>:	pop-up menu(vtMenu) ¥n¥
~Ctrl ~Meta <Btn2Down>:	ignore ¥n¥
~Ctrl ~Meta <Btn2Up>:	insert-selection(PRIMARY, CUT_BUFFER0) ¥n¥
!Ctrl <Btn3Down>:	pop-up menu(fontMenu) ¥n¥
!Lock Ctrl <Btn3Down>:	pop-up menu(fontMenu) ¥n¥
~Ctrl ~Meta <Btn3Down>:	start-extend ¥n¥
~Meta <Btn3Motion>:	select-extend ¥n¥
<BtnUp>:	select-end(PRIMARY, CUT_BUFFER0) ¥n¥
<BtnDown>:	bell(0)

Tektronix ウィンドウにおけるデフォルトの割り当ては、下記のとおりです。

~Meta<KeyPress>:	insert-seven-bit ¥n¥
Meta<KeyPress>:	insert-eight-bit ¥n¥
!Ctrl <Btn1Down>:	pop-up menu(mainMenu) ¥n¥
!Lock Ctrl <Btn1Down>:	pop-up menu(mainMenu) ¥n¥
!Ctrl <Btn2Down>:	pop-up menu(tekMenu) ¥n¥
!Lock Ctrl <Btn2Down>:	pop-up menu(tekMenu) ¥n¥
Shift ~Meta<Btn1Down>:	gin-press(L) ¥n¥
~Meta<Btn1Down>:	gin-press(l) ¥n¥
Shift ~Meta<Btn2Down>:	gin-press(M) ¥n¥
~Meta<Btn2Down>:	gin-press(m) ¥n¥
Shift ~Meta<Btn3Down>:	gin-press(R) ¥n¥
~Meta<Btn3Down>:	gin-press(r)

下記の例は、**keymap** アクションを使って、一般的なタイプ作業入力用の特殊キーを追加する例です。

```
*VT100.Translations:          #override <Key>F13: keymap(dbx)
*VT100.dbxKeymap.translations:
¥

<Key>F14:    keymap(None) ¥n¥
<Key>F17:    string("next") string(0x0d) ¥n¥
<Key>F18:    string("step") string(0x0d) ¥n¥
<Key>F19:    string("continue") string(0x0d) ¥n¥
<Key>F20:    string("print ")
              insert-selection(PRIMARY,CUT_BUFFER0)
```

環境

xterm コマンドは、ユーザーが作成したウィンドウのサイズに、適切な **TERM** および **TERMCAP** 環境変数を設定します。また、**DISPLAY** 環境変数を使用または設定して、使用するビットマップ・ディスプレイ端末を指定します。**WINDOWID** 環境変数は、**xterm** ウィンドウの X のウィンドウ ID 番号に設定されます。

バグ

システムによっては、大きなペーストが動作しない場合があります。これは **xterm** コマンドのバグではなく、そのシステムの疑似ターミナル・ドライバのバグです。**xterm** コマンドが疑似ターミナルに送ることができるペーストの大きさは、疑似ターミナルがデータを受け取る速さによりますが、書き込み操作が成功したときにその旨を示す十分な情報を戻さない疑似ターミナル・ドライバもあります。

多くのオプションは、**xterm** コマンドの始動後はリセットできません。

固定幅で文字セルのフォントのみがサポートされています。

制御シーケンス

このセクションでは、**xterm** コマンドで使用可能な制御シーケンスをリストします。

定義

下記の情報は、このセクションでのキー・シーケンスを解釈する方法を示しています。

項目	説明
<i>c</i>	リテラル文字 <i>c</i>
<i>C</i>	単一 (必須) 文字
<i>P_s</i>	1 つ以上の数字からなる、単一の数値パラメーター (通常はオプション)
<i>P_m</i>	; (セミコロン) で区切られた数個の単一数値パラメーターからなる複数数値パラメーター。
<i>P_t</i>	印刷可能文字からなるテキスト・パラメーター

VT100 モード

このセクションの制御シーケンスのほとんどは、標準 VT102 制御シーケンスですが、それ以降の DEC VT 端末からの制御シーケンスもあります。サポートされていない VT102 機能の主なものは低速スクロール、縦倍角文字、明滅文字、および VT52 モードです。

また、**xterm** に依存する機能 (スクロール・バー、ウィンドウ・サイズなど) を提供するために追加された制御シーケンスがあります。DEC または ISO 6429 で指定されている機能については、その機能に割り当てられているコードが括弧で囲んであります。文字セットを指定するエスケープ・コードは、ISO 2022 で

指定されています。文字セットについては、ISO 2022 関連の資料を参照してください。

制御シーケンス	説明
BEL	ベル (Ctrl+G)
BS	バックスペース (Ctrl+H)
TAB	水平タブ (HT) (Ctrl+l)
LF	改行 (LF) または改行 (NL) (Ctrl+J)
VT	LF と同じ垂直タブ (Ctrl+K)
FF	LF と同じ改ページまたは改ページ (NP) (Ctrl+L)
CR	復帰 (Ctrl+M)
SO	シフトアウト (Ctrl+N) -> 代替文字セットへの切り替え: G1 文字セットを起動
SI	シフトイン (Ctrl+O) -> 標準文字セットへの切り替え: G0 文字セットを起動 (デフォルト)
ESC # 8	DEC 画面検査 (DCECALN)
ESC (C	G0 文字セット (ISO 2022) を指定 C = 0 DEC 特殊文字および線画セット C = A グレートブリテンおよび北部アイルランド連合王国 (UK) C = B アメリカ合衆国 (USASCII)
ESC) C	G1 文字セット (ISO 2022) を指定 C = 0 DEC 特殊文字および線画セット C = A グレートブリテンおよび北部アイルランド連合王国 (UK) C = B アメリカ合衆国 (USASCII)
ESC * C	G2 文字セット (ISO 2022) を指定 C = 0 DEC 特殊文字および線画セット C = A グレートブリテンおよび北部アイルランド連合王国 (UK) C = B アメリカ合衆国 (USASCII)
ESC + C	G3 文字セット (ISO 2022) を指定 C = 0 DEC 特殊文字および線画セット C = A グレートブリテンおよび北部アイルランド連合王国 (UK) C = B アメリカ合衆国 (USASCII)
ESC 7	カーソルを保存 (DECSC)
ESC 8	カーソルを復元 (DECRC)
ESC =	アプリケーション・キーパッド (DECPAM)
ESC >	標準キーパッド (DECNM)
ESC D	索引 (IND)
ESC E	次行 (NEL)
ESC H	タブ・セット (HTS)
ESC M	逆インデクシング (RI)
ESC N	G2 文字セット (SS2) のシングル・シフト選択: 次の文字にだけ影響
ESC P	G3 文字セット (SS2) のシングル・シフト選択: 次の文字にだけ影響。
ESC O P_t ESC ¥	装置制御文字列 (DCS)。 xterm は DCS 機能をインプリメントしません; P _t は無視されます。 P _t は印刷可能文字である必要はありません。
ESC Z	端末装置識別機構 (DECID) を戻す。 ESC [c (DA) の廃止形式です。
ESC [P_s @	文字の P _s (ブランク) 文字 (デフォルト=1) を挿入 (ICH)
ESC [P_s A	P _s 回 (デフォルト=1) カーソル・アップ (CUU)

制御シーケンス	説明
ESC [P_s B	P_s 回 (デフォルト=1) カーソル・ダウン (CUD)
ESC [P_s C	P_s 回 (デフォルト=1) カーソル・フォワード (CUF)。

項目	説明
ESC [P_s D	P_s 回 (デフォルト=1) (CUB) カーソル・バックワード。
ESC [P_s ; P_s H	カーソル位置 [行; 桁] (デフォルト=1) (CUP)
ESC [P_s J	ディスプレイ内の消去 (ED) $P_s = 0$ 下を消去 (デフォルト) $P_s = 1$ 上を消去 $P_s = 2$ 全消去
ESC [P_s K	行の消去 (EL) $P_s = 0$ 右に向かって消去 (デフォルト) $P_s = 1$ 左に向かって消去 $P_s = 2$ 全消去
ESC [P_s L	P_s 行 (デフォルト=1) 追加 (IL)
ESC [P_s M	P_s 行 (デフォルト=1) 削除 (DL)
ESC [P_s P	P_s 文字 (デフォルト=1) 削除 (DCH)
ESC [P_s ; P_s ; P_s ; P_s ; P_s T	hilit mouse トラッキングを開始。パラメーターは、 [Func;Startx;Starty;FirstRow;LastRow] です。 マウス・トラッキングを参照してください。
ESC [P_s c	SendDevice 属性 (DA) が P_s 文字 (デフォルト=1) 削除 (DCH)。 $P_s = 0$ または省略 端末からの要求属性 ESC [? 1 ; 2 c ("I am a VT100 with Advanced Video Option.")
ESC [P_s ; P_s f	水平または垂直位置 [行; 桁] (デフォルト= [1,1]) (HVP)
ESC [P_s g	タブ消去 (TBC) $P_s = 0$ 現在の桁を消去 (デフォルト) $P_s = 3$ 全消去
ESC [P_m h	設定モード (SM) $P_s = 4$ 挿入モード (IRM) $P_s = 2 0$ 自動改行 (LNM)
ESC [P_m l	リセット・モード (RM) $P_s = 4$ 置換モード (IRM) $P_s = 2 0$ 標準改行 (LNM)

項目	説明
ESC [P_m m	文字属性 (SGR) $P_s = 0$ Normal (デフォルト) $P_s = 1$ 太字 $P_s = 4$ 下線 $P_s = 5$ 明滅 (太字で表示) $P_s = 7$ 反転
ESC [P_s n	デバイス状況レポート (DSR) $P_s = 5$ 状況報告ESC [0 n ("OK") $P_s = 6$ カーソル位置 [行; 桁] を次のとおり報告 (CPR)ESC [r ; c R $P_s = 2 0$ 自動改行 (LNM)
ESC [P_s ; P_s r	スクロール領域 [トップ; ボトム] (デフォルト= ウィンドウのフルサイズ) を設定 (DECSTBM)
ESC [P_s x	端末パラメーターを要求 (DECREQTPARM)

項目	説明
ESC [? <i>P_s</i> h	<p>DEC プライベート・モード (DECSET)</p> <p><i>P_s</i> = 1 アプリケーション・カーソル・キー (DECCKM)</p> <p><i>P_s</i> = 2 文字セット G0-G3 に対して USASCII を指定。(VT102 では、これは、xterm がサポートしていない VT52 モード (DECANM) を選択します。)</p> <p><i>P_s</i> = 3 132 桁モード (DECCOLM)</p> <p><i>P_s</i> = 4 低速スクロール (DECSCLM)</p> <p><i>P_s</i> = 5 反転表示 (DECSCNM)</p> <p><i>P_s</i> = 6 オリジン・モード (DECOM)</p> <p><i>P_s</i> = 7 循環モード (DECAWM)</p> <p><i>P_s</i> = 8 オートリピート・キー (DECARM)</p> <p><i>P_s</i> = 9 ボタンを押すときにマウス X と Y を設定する。 267 ページの『マウス・トラッキング』を参照してください。</p> <p><i>P_s</i> = 3 8 Tektronix モードを入力 (DECTEK)</p> <p><i>P_s</i> = 4 0 80 <—> 132 モードを許可する</p> <p><i>P_s</i> = 4 1 cursor 機能固定</p> <p><i>P_s</i> = 4 4 マージン・ベルをオンにする</p> <p><i>P_s</i> = 4 5 逆循環モード</p> <p><i>P_s</i> = 4 7 代替画面バッファを使用 (titelnhibit リソースによって使用不可にされていない場合)</p> <p><i>P_s</i> = 1 0 0 0 ボタンを押して放すときにマウス X と Y を設定する。 267 ページの『マウス・トラッキング』を参照してください。</p> <p><i>P_s</i> = 1 0 0 1 Hilite マウス・トラッキングを使用。</p>

項目	説明
ESC [? P_m 1	DEC プライベート・モード・リセット (DECRST) $P_s = 1$ 標準カーソル・キー (DECCKM) $P_s = 3$ 80 桁モード (DECCOLM) $P_s = 4$ ジャンプ高速スクロール (DECSCLM) $P_s = 5$ 標準ビデオ (DECSCNM) $P_s = 6$ 標準カーソル・モード (DECOM) $P_s = 7$ 非循環モード (DECAWM) $P_s = 8$ 非オートリピート・キー (DECARM) $P_s = 9$ ボタンを押すときにマウス X と Y を送信しない $P_s = 4 0$ 80 <—> 132 を不許可にする $P_s = 4 1$ 非 curses 機能固定 $P_s = 4 4$ マージン・ベルをオフにする $P_s = 4 5$ 非逆循環モード $P_s = 4 7$ 標準画面バッファを使用 $P_s = 1 0 0 0$ ボタンを押すときと放すときにマウス X および Y を送信しない $P_s = 1 0 0 1$ Hilite マウス・トラッキングを使用しない。xxx
ESC [? P_m r	DEC プライベート・モード値を復元する。以前に保存された P_s の値が復元されます。 P_s 値は DECSET と同じです。
ESC [? P_m s	DEC プライベート・モード値を保存する。 P_s 値は DECSET と同じです。
ESC]? P_s ; P_t BEL	テキスト・パラメーターを設定 $P_s = 0$ アイコン名とウィンドウ・タイトルを P_t に変更 $P_s = 1$ アイコン名を P_t に変更 $P_s = 2$ ウィンドウ・タイトルを P_t に変更 $P_s = 5 0$ フォントを P_t に設定
ESC P_t ESC ¥	プライベート・メッセージ (PM)。 xterm は PM 機能をインプリメントしません; P_t は印刷可能文字である必要はありません。
ESC _ P_t ESC ¥	アプリケーション・プログラム・コマンド (APC)。プライベート・メッセージ (PM)。 xterm は APC 機能をインプリメントしません; P_t は無視されます。 P_t は印刷可能文字である必要はありません。
ESC c	フル・リセット (RIS)
ESC n	G2 文字セットを選択 (LS2)
ESC o	G3 文字セットを選択 (LS3)
ESC l	GR として G3 文字セットを起動 (LS3R)。 xterm での可視効果はありません。

項目	説明
ESC }	GR として G2 文字セットを呼び出す (LS2R)。 xterm での可視効果はありません。
ESC	GR として G1 文字セットを呼び出す (LS1R)。 xterm での可視効果はありません。

XTERM 記述制約

AIX バージョン 4 の DEC.TI ファイルにおける **xterm** 端末記述により、SGR 属性の使用による下線モードが提供されます。SMUL および RMUL 属性は、現在 AIX バージョン 4 の XTERM 端末記述に定義されていません。より汎用的な SGR という機能を使用します。

```
tput sgr x y
```

ここで、*x* は 1 または 0 で、それぞれスタンドアウト・モードをオンまたはオフにします。*y* は 1 または 0 で、それぞれ下線モードをオンまたはオフにします。SGR 機能についての詳細は、**terminfo** ファイル・フォーマットのセクションを参照してください。

```
tput sgr 0 1    turn off standout; turn on underline
tput sgr 0 0    turn off standout; turn off underline
tput sgr 1 1    turn on standout; turn on underline
tput sgr 1 0    turn on standout; turn off underline
```

マウス・トラッキング

マウス・ボタンを押したときにマウスの位置や他の情報を送信するように **VT** ウィジェットを設定することができます。これらのモードは、通常、マウスを使用するエディターなどのフルスクリーン・アプリケーションで使用されます。

モードは 3 つあり、これらを同時に使用することはできません。それぞれのモードは DECSET (または DECST) エスケープ・シーケンスの中の異なるパラメーターを使用することで切り替えられます。**xterm** コマンドが生成するすべてのマウス追跡エスケープ・シーケンスに対するパラメーターは、*value+040* の形で、常に単一文字の数値パラメーターにエンコードされます。画面座標システムのベースは 1 です。

例えば、! は 1 で、画面座標システムのベースは 1 です。

X10 互換モードでは、マウスのボタンを押したとき、マウスを押した位置とそのボタンをエンコードしてからエスケープ・シーケンスを送信します。これは、DECSET にパラメーター 9 を指定することで使用可能になります。ボタンを押したとき、**xterm** コマンドは下記の「6 文字」を送信します。Cb はボタン 1 です。Cx および Cy は、ボタンを押したときのマウスの *x* 座標と *y* 座標です。

```
ESC [ M CbCxCy
```

通常の追跡モードでは、ボタンを押したときと離れたときの両方でエスケープ・シーケンスが送信されます。修飾子情報も送信されます。これは、DECSET にパラメーター 1000 を指定することで使用可能になります。ボタンを押したとき、または離れたとき、**xterm** コマンドは下記の「キー・シーケンス」を送信します。

```
ESC [ M CbCxCy
```

C_b の下位 2 ビットは、ボタン情報をエンコードします。それぞれの意味は、0=MB1 が押された、1=MB2 が押された、2=MB3 が押された、3=解放された、ということです。上位ビットは、ボタンが押されたと

きに押し下げられ、一緒に追加された修飾子をエンコードします。それぞれの意味は、4 = シフト、8 = メタ、16 = 制御です。 C_x および C_y は、マウス・イベントの x 座標と y 座標です。左上端は (1, 1) です。

マウス・ハイライト追跡は、まず、ボタンが押されたことをプログラムに通知し、ある範囲の行をプログラムから受信し、ボタンが離されるまでその範囲内でマウスがカバーしている領域を強調表示し、最後に、ボタンが離された座標をプログラムに送信します。これは、DECSET にパラメーター 1001 を指定することで使用可能になります。

重要: このモードを使用するには共同プログラムが必要です。このプログラムがなければ、**xterm** コマンドはハングしてしまいます。ボタンが押されたとき、通常の追跡と同じ情報が生成されます。そして、**xterm** コマンドはプログラムからマウス追跡情報が送られてくるのを待ちます。下記の適切なエスケープ・シーケンス が疑似ターミナルから送信されてこなければ、すべての X イベントは無視されてしまいます。

ESC [$P_s ; P_s ; P_s ; P_s ; T$

パラメーターは、*Func*、*Startx*、*Starty*、*FirstRow*、*LastRow* です。*Func* パラメーターは、ゼロ以外の値の場合、ハイライト追跡を開始し、0 (ゼロ) の場合、ハイライト追跡を異常終了します。*Startx* および *Starty* パラメーターは、ハイライトされた領域の開始 x 座標および y 座標を意味します。終了位置でマウスを追跡しますが、*FirstRow* 行より上になることはなく、常に *LastRow* 行より上になります (画面の最上部は第 1 行です)。ボタンを放すと、**xterm** コマンドは終了位置を下記の 2 つのうちのいずれかの方法で報告します。開始座標と終了座標が有効なテキストの位置にある場合、**xterm** コマンドは「終了位置」を下記のように報告します。

ESC [t $C_x C_y$

どちらかの座標が行の終わりを超えると、**xterm** コマンドは "終了位置" を下記のように報告します。

ESC [T $C_x C_y C_x C_y C_x C_y$

パラメーターは、*Startx*、*Starty*、*Endx*、*Endy*、*Mousex*、*Mousey* です。*Startx*、*Starty*、*Endx*、*Endy* パラメーターは、領域の開始する文字位置および終了する文字位置を与えます。*Mousex* および *Mousey* パラメーターは、ボタンを離したときのマウスの位置を与えます。これらの位置データの値が、1 文字分の数値の大きさを超えることはありません。

Tektronix 4014 モード

このセクションのシーケンスのほとんどは、標準 Tektronix 4014 制御シーケンスです。使用できない主な機能は、ライト・スルー (write-thru) モードとフォーカス解除 (defocused) モードです。ここでは、さまざまな Tektronix プロット・モードで使用されるコマンドは説明せず、モードを切り替えるコマンドについて説明します。

関連資料:

14 ページの『vi または vedit コマンド』

関連情報:

aixterm コマンド

resize コマンド

xwd コマンド

目的

拡張 X Window システム のウィンドウ・イメージをダンプします。

構文

```
xwd [ -add Value ] [ -frame ] [ -display Display ] [ -help ] [ -nobdrs ] [ -xy ] [ -out File ]  
[ -root | -id id | -name Name ] [ -icmap ] [ -screen ]
```

説明

xwd コマンドは、拡張 X Window システム のウィンドウ・ダンプ作成ユーティリティです。**xwd** コマンドを使えば、ウィンドウ・イメージを特別にフォーマット済みダンプ・ファイルにして保管することができます。このファイルは、後で他のさまざまな X ユーティリティから読み取って、再表示、印刷、編集、フォーマット、アーカイブ、イメージ処理などの作業を行うことができます。ターゲット・ウィンドウは、そのウィンドウ内でマウスをクリックすることによって選択します。ダンプの開始時に 1 回、またダンプの完了時に 2 回、それぞれキーボード・ベルが鳴ります。

フラグ

項目	説明
-add Value	すべてのピクセルに追加される符号付きの値を 1 つ指定します。このオプションは X11R5 に固有なものです。
-frame	このオプションは、ウィンドウを手操作で選択しているとき、ウィンドウ・マネージャー・フレームが含まれていなければならないことを示します。
-display Display	接続するサーバーを指定します。
-help	使用方法とコマンド構文の要約を表示します。
-nobdrs	X Window のボーダーを構成するピクセルをウィンドウ・ダンプに入れないことを指定します。これは、ウィンドウの内容をイラストとして文書内に入れたいときに有効です。 -nobdrs フラグの結果は、どのウィンドウ・マネージャーが稼働しているかによって変わります。多くのウィンドウ・マネージャーはクライアントからすべてのボーダーを除去します。例えば、 XGetWindowAttributes 関数は、クライアントの始動時に、ボーダー幅にかかわらず、 border_width フィールドに対して 0 の値を戻します。そのため、画面上に表示されるボーダーはいずれもウィンドウ・マネージャーに属し、クライアントにはその情報が知らされません。この場合、 -nobdrs フラグは効果がありません。
-out File	コマンド・ラインで出力ファイルを指定します。デフォルトは標準出力への出力です。
-root	ユーザーがポインターを用いてウィンドウを選択しなくても、ルート・ウィンドウがウィンドウ・ダンプのために選択されることを示します。このオプションは X11R5 に固有なものです。
-id id	ユーザーがポインターを用いてウィンドウを選択しなくても、指定されたリソース ID を持つウィンドウがウィンドウ・ダンプのために選択されることを示します。このオプションは X11R5 に固有なものです。
-name Name	ユーザーがポインターを用いてウィンドウを選択しなくても、指定された WM_NAME 属性を持つウィンドウがウィンドウ・ダンプのために選択されることを示します。このオプションは X11R5 に固有なものです。
-icmap	最初にインストールされた画面のカラー・マップを強制的に使って、RGB 値を獲得します。デフォルトでは、選択されたウィンドウのカラー・マップが使用されます。このオプションは X11R5 に固有なものです。
-screen	イメージを得るために使用される getImage 要求が指定されたウィンドウ上で直接行われるのではなく、ルート・ウィンドウ上で行われることを指定します。この方法によって、ユーザーは指定されたウィンドウに重なる他のウィンドウをいくつか獲得することができますが、さらに重要なのは、メニューやその他のポップアップなど独立したウィンドウを指定されたウィンドウの上にかぶせて見ることができる点です。このオプションは X11R5 に固有なものです。
-xy	デフォルトの z フォーマットの代わりに xy フォーマットのダンプを選択します。このオプションはカラー・ディスプレイだけに適用されます。

ファイル

項目	説明
XWDFile.h	X Window のダンプ・ファイル・フォーマット定義ファイル。

xwud コマンド

目的

拡張 X-Window のウィンドウのダンプ・イメージを取り出して表示します。

構文

```
xwud [ -in FileName ] [ -noclick ] [ -geometry Geometry ] [ -display Display ] [ -new ] [ -std MapType ] [ -raw ] [ -vis visual_type | visual_id ] [ -help ] [ -rv ] [ -plane Number ] [ -fg Color ] [ -bg Color ]
```

説明

xwud コマンドは、拡張 X-Window のウィンドウ・ダンプ・イメージを取り出します。そのために、以前に **xwd** コマンドで生成された特殊フォーマットのダンプ・ファイルに保管されていたイメージを、ウィンドウに表示します。ダンプ・ファイルのフォーマットは、**XWDFile.h** ファイルによって決まります。

フラグを使用して、カラー・ディスプレイ、ウィンドウのサイズと位置、入力フィールド、およびビジュアル・クラスまたは ID を指定することができます。また、表示したいイメージの単一ビット平面を選択することもできます。

フラグ

項目	説明
-bg Color	ビットマップ・イメージ (またはイメージの単一平面) が表示される場合に、イメージ内の 0 (ゼロ) のビットに表示する色を指定します。
-display Display	接続先となるサーバーを指定します。 X コマンドのセクションを参照してください。
-fg Color	ビットマップ・イメージ (またはイメージの単一平面) が表示される場合に、イメージ内の 1 のビットに表示する色を指定します。
-geometry Geometry	ウィンドウのサイズと位置を指定します。通常は位置のみを指定し、サイズはデフォルト値をイメージの実際のサイズにしておきます。
-help	有効なオプションの簡単な説明を表示します。
-in FileName	コマンド・ラインで入力ファイルを指定します。入力ファイルを指定しなければ、標準入力 that 想定されます。
-new	イメージ表示用の新しいカラー・マップを作成します。イメージ特性がディスプレイの特性と一致する場合は、このフラグを使用するとイメージの画面表示速度が高速になりますが、新しいカラー・マップを使用しなければなりません (ほとんどの端末では、他のウィンドウがテクニカラーになります)。
-noclick	ウィンドウ内のボタンをクリックしたときに、アプリケーションが終了しないようにします。文字 q または Q あるいは Ctrl-C キー・シーケンスを入力すると、アプリケーションを終了することができます。
-plane Number	表示したいイメージの単一ビット平面を選択します。平面には番号が付いており、最下位有効ビットが 0 (ゼロ) です。このフラグを使用して、どの平面を表示用に xpr コマンドに渡すかを決定します。

項目	説明
-raw	スクリーン上に現在存在するカラー値でダンプ・イメージを表示します。このフラグは、元のウィンドウが引き続き表示されている間に、最初に表示した同じ画面にイメージを逆ダンプするときに便利です。これにより、イメージの画面表示が高速になります。
-rv	ビットマップ・イメージ (またはイメージの単一平面) を表示する場合のフォアグラウンド・カラーとバックグラウンド・カラーをスワップします。このフラグは、ピクセル値が 0 と 1 のカラー・センスがディスプレイ上で反転されているビットマップ・イメージを表示するときに便利です。
-std MapType	指定された標準カラー・マップを使用してイメージを表示します。タイプを大文字に変換し、先行 RGB_ と後続 _MAP を追加すると、マップ・タイプを得ることができます。一般的なマップ・タイプは best 、 default 、 gray です。標準カラー・マップの作成については、 /usr/lpp/X11/Xamples/clients/xstdcmap を参照してください。
-vis visual_type visual_id	特定のビジュアル・タイプまたはビジュアル ID を指定します。デフォルトでは、 best のタイプを取り出すか、または default を指定できます。また、 default を指定すると、ルート・ウィンドウのカラー・マップと同じクラスになります。

StaticGray、**GrayScale**、**StaticColor**、**PseudoColor**、**DirectColor**、**TrueColor** の中から、特定のクラスを指定することができます。ソース・イメージと同じクラスを使用するには、**Match** を指定します。

正確なビジュアル ID (サーバーに特定の ID) を、16 進数 (先行 0x 付き) または 10 進数として指定してください。この文字列では、大文字と小文字は区別されません。

環境変数

項目	説明
DISPLAY	デフォルト・ディスプレイを取り出します。

例

ダンプ・ウィンドウから特定のファイルを取り出すには、下記のように入力します。

```
xwud -in FileName
```

関連資料:

224 ページの『xpr コマンド』

269 ページの『xwd コマンド』

y

以下の AIX コマンドは、文字 *y* で始まります。

yacc コマンド

目的

コンテキストのない文法仕様からなる入力データから、LALR(1) 解析プログラムを生成します。

構文

```
yacc [ -b Prefix ] [ -C ] [ -d ] [ -l ] [ -NnNumber ] [ -NmNumber ] [ -NrNumber ] [ -p Prefix ]  
[ -s ] [ -t ] [ -v ] [ -y Path ] Grammar
```

説明

yacc コマンドは、コンテキストのない文法仕様を、LALR(1) 解析アルゴリズムを実行する単純なオートマトン用テーブル・セットに変換します。文法があいまいなので、そのために優先順位の規則を使用して、あいまいな問題点を解決します。

関数 **yyparse** を生成するには、C 言語コンパイラを使って出力ファイル **y.tab.c** をコンパイルしなければなりません。この関数は字句解析プログラム **yylex** の他に **main** サブルーチンや、**yyerror** エラー処理 サブルーチンと一緒にロードしなければなりません (これらのサブルーチンはユーザーが提供しなければなりません)。 **yyparse** サブルーチンが使うことのできる字句解析プログラムの作成には **lex** コマンドが便利です。 **main** および **yyerror** サブルーチンの簡略バージョンは、**yacc** ライブラリー、**liby.a** を通じて使用することができます。また、**yacc** を使用して C++ 出力も生成できます。

yacc で生成された C ファイル (**y.tab.c**) を **-DYACC_MSG** オプションでコンパイルすることで、メッセージ機能を使用するために必要なコードを含めることができます。コンパイル中にこのオプションを使用すると、**yyparse** サブルーチンおよび **YYBACKUP** マクロによって生成されたエラー・メッセージは、**yacc_user.cat** カタログから抽出されます。

これによって、ユーザーは、英語圏でない地域において、英語以外の言語でエラー・メッセージを受信することができます。カタログが見つからなかった場合、もしくはオープンできなかった場合には、**yyparse** および **YYBACKUP** サブルーチンは、デフォルトの英語メッセージを表示します。

yacc コマンドは、**LANG**、**LC_ALL**、**LC_CTYPE**、**LC_MESSAGES** 環境変数によって影響を受けます。

フラグ

項目	説明
-b Prefix	すべての出力ファイル名の接頭部に y の代わりに <i>Prefix</i> を使用します。コード・ファイル y.tab.c 、ヘッダー・ファイル y.tab.h (-d が指定されたときに作成されるファイル) および記述ファイル y.output (-v が指定されたときに作成されるファイル) は、それぞれ <i>Prefix.tab.c</i> 、 <i>Prefix.tab.h</i> 、 <i>Prefix.output</i> に変更されます。
-C	C++ コンパイラーで使用するために、 y.tab.c ファイルの代わりに y.tab.C ファイルを生成します。入出力用の入出力ストリーム・ライブラリーを使用するには、マクロ _CPP_IOSTREAMS を定義します。
-d	ファイル y.tab.h を生成します。このファイルには、 yacc 割り当てトークン・コードをユーザーのトークン名と関連付ける #define ステートメントが含まれます。これによって y.tab.c 以外のソース・ファイルが、このヘッダー・ファイルを含めることによってトークン・コードへアクセスできるようになります。
-l	y.tab.c の中に #line 構成を含めません。これは文法およびそれに対応する作業が完全にデバッグされてから使ってください。
-NnNumber	トークンと非端末名の配列のサイズを <i>Number</i> に変更します。デフォルト値は 8000 です。有効値は、8000 より大きいもののみです。
-NmNumber	メモリー状態の配列のサイズを <i>Number</i> に変更します。デフォルト値は 40000 です。有効値は、40000 より大きいもののみです。
-NrNumber	大量の文法を処理するために、内部バッファのサイズを変更します。デフォルト値は 2000 です。有効な値は、2000 よりも大きいもののみです。
-p Prefix	yacc コマンドで作成されるすべての外部名の接頭部として、 yy の代わりに <i>Prefix</i> を使用します。影響を受ける外部名は、 yychar 、 yyval 、 yydebug 、 yyparse() 、 yylex() 、 yyerror() です (以前は、 -p を使用して代替パーサーを指定していましたが、現在では -yPath を使用して代替パーサーを指定することができます)。
-s	yyparse 関数を複数の小さな関数に分割します。そのサイズは文法の大きさにある程度比例するため、 yyparse 関数はコンパイル、最適化、あるいは効率的実行に支障をきたすほど大きくなってしまいますことがあります。
-t	実行時デバッグ・コードをコンパイルします。デフォルトでは、 y.tab.c がコンパイルされるときこのコードは含まれません。ただし、実行時デバッグ・コードはプリプロセッサ・マクロ YYDEBUG に制御されます。 YYDEBUG が 0 以外の値をとるとき、 -t フラグが使われているかどうかに関係なく、C コンパイラー (cc) にはデバッグ・コードが含まれます。コンパイラーにデバッグ・コードを含めたくない場合には、 YYDEBUG は 0 の値をとらなければなりません。このコードをコンパイルしなければ、 yyparse サブルーチンは稼働速度が速くなります。
	-t フラグを指定すると、デバッグ・コードがコンパイルされますが、実際にデバッグ・モードがオンになるわけではありません。デバッグ出力を表示するには、 yacc 文法ファイルの宣言セクションに C 言語宣言 <code>int yydebug=1</code> を追加するか、または dbx を通じて yydebug を設定して、 yydebug 変数を設定しなければなりません。
-v	ファイル y.output を準備します。このファイルには読み取り可能な構文解析テーブルの記述と、あいまいな文法から生成された矛盾についての報告が含まれます。
-y Path	<i>Path</i> で指定したパーサーのプロトタイプを、デフォルトの /usr/lib/yaccpar ファイルの代わりに使用します。(以前は、代替パーサーを指定するために -p を使用していました。)

終了状況

このコマンドは、下記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. 下記のコマンドは、

```
yacc grammar.y
```

ファイル **grammar.y** から **yacc** 規則を取り出し、出力を **y.tab.c** に入れます。

2. 下記のコマンドは、

```
yacc -d grammar.y
```

例 1 と同様に機能しますが、**y.tab.h** ファイルも生成します。このファイルには、**grammar.y** ファイル内で定義したトークンごとに、C スタイルの **#define** ステートメントが入っています。

ファイル

項目	説明
y.output	構文解析テーブルの読み取り可能な記述、およびあいまいな文法によって生成された矛盾の報告が入っています。
y.tab.c	出力ファイルが入っています。
y.tab.h	トークン名の定義が入っています。
yacc.tmp	一時ファイル。
yacc.debug	一時ファイル。
yacc.acts	一時ファイル。
/usr/ccs/lib/yaccpar	C プログラム用パーサーのプロトタイプが入っています。
/usr/ccs/lib/liby.a	実行時ライブラリーが入っています。

関連情報:

lex コマンド

lex コマンドを使用しての字句解析プログラムの生成

lex および yacc プログラムの例のセクション

yes コマンド

目的

肯定の応答を繰り返し出力します。

構文

yes [*charstring*]

説明

yes コマンドは、肯定応答を繰り返し出力します。**yes** コマンドは、指定されたアクションを完了する前に肯定応答を必要とする別のコマンドへのパイプ入力として使用してください。例えば、**yes** コマンドは、ディレクトリーから複数のファイルを削除する場合に便利です。Ctrl-C キー・シーケンスを入力すると、連続する肯定応答が終了します。

注: 現行ロケールは、**LC_MESSAGES** 環境変数、または、指定されていれば *charstring* パラメーターによって決定されます。*charstring* パラメーターには、任意の単一文字または文字ストリームを使用することができます。**yes** コマンドを発行した後で *charstring* パラメーターを入力すると、Ctrl-C キー・シーケンスを入力するまで、*charstring* パラメーターが画面に表示されます。

例

ワード **first** を画面に表示するには、下記のように入力します。

```
yes first
```

このステートメントにより、Ctrl-C キー・シーケンスを入力するまで、ワード **first** が表示されます。

ファイル

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

関連情報:
環境ファイル

ypbind デーモン

目的

クライアント・プロセスが NIS サーバーとバインド、すなわち接続できるようにします。

構文

```
/usr/lib/netsvc/yp/ypbind [ -s -ypset -ypsetme ]
```

説明

ypbind デーモンは、Network Information Services (NIS) クライアント上のプロセスを、NIS サーバー上のサービスにバインド (つまり接続) します。このデーモンはすべての NIS クライアント上で稼働し、下記のシステム・リソース・コントローラー (SRC) コマンドによって開始または終了されます。

```
startsrc -s ypbind
```

```
stopsrc -s ypbind
```

Network Information Services (NIS) マップからクライアントが情報を要求すると、**ypbind** デーモンはサーバーに対してネットワーク上で情報をブロードキャストします。サーバーが応答すると、サーバーがデーモンにホストの IP アドレスとポート番号を与えます。これが、クライアントが探している情報を提供するホストです。**ypbind** デーモンはこのアドレス情報を、**domainname.version** というファイル名を使用して `/var/yp/binding` ディレクトリー内へ保管します。それから、クライアントが次に NIS マップにアクセスしようとしたときに、クライアントの **ypbind** デーモンは **domainname.version** ファイル内のアドレスを参照します。

ypbind デーモンは、複数のドメインとそのサーバー **-ypsetme** へのバインドを同時に保守することができます。デフォルトのドメインは、始動時に **domainname** コマンドによって指定されたドメインです。

注:

1. ドメインがアンバインドされたとき (通常はサーバーが壊れたか過負荷になったとき) は、**ypbind** デーモンは他のサーバーを探して再度情報をブロードキャストします。
2. クライアントを特定のサーバーに強制的にバインドするには、**ypset** コマンドを使います。
3. クライアントがどのサーバーにバインドされるかを知るには、**ypwhich** コマンドを使います。
4. `/var/yp/binding/domainname/ypservers` ファイルが既に存在する場合は、**ypbind** は、ブロードキャストする前に、そのファイル内にリストされたサーバーへ連絡しようとします。ファイルには、サーバーの IP アドレスを一行に 1 つずつリストしたものが入っている必要があります。
5. デフォルトでは、NIS クライアントは NIS サーバーからの応答を無期限に待ちます。その間はクライアント・システムにログインできません。ただし、この待ち時間の長さを制限することは可能です。**ypbind** デーモンを開始する前に、**YPBIND_MAXWAIT** 環境変数を (通常は `/etc/environment` で) 設定しますが、この値 (秒単位) により NIS クライアントが NIS サーバーを待つ時間が制限されます。この制限時間を超えると、クライアントは NIS が使用不能であるとして、ローカル・ファイルの使用を継続します。この場合、例えば `root` は、ローカルでのログインを許可されます。

6. ドメインがアンバインドされ `/var/yp/binding/domainname/ypservers` ファイル内にリストされた場合、デフォルトでは `ypbind` デーモンは現在停止しているサーバーに連絡しようとしています。しかし、`ypbind daemon` を開始する前に `YPBIND_SKIP` 環境変数が 1 に設定されている場合 (通常 `/etc/environment` ファイルに設定されています)、現在停止しているサーバーに再び連絡することはありません。

フラグ

項目	説明
<code>-s</code>	特権を持つ通信ポート上で <code>ypbind</code> デーモンをセキュア・モードで実行します。
<code>-ypset</code>	ローカル・ホストが、ローカル・ホストまたはリモート・ホストからの <code>ypset</code> コマンドを受け入れることを示します。
<code>-ypsetme</code>	ローカル・ホストが、ローカル・ホストからの <code>ypset</code> コマンドのみを受け入れることを示します。このフラグを <code>-ypset</code> フラグとともに指定すると、 <code>-ypset</code> フラグが上書きされます。

注:

1. `-ypset` フラグも `-ypsetme` フラグも指定しなければ、ローカル・ホストは全ホストからの `ypset` コマンドをすべて拒否します。NIS サーバーが変更できないため、これは最も厳密に保護されたモードです。
2. `-ypset` フラグも `-ypsetme` フラグも指定しなければ、ローカル・ホストは全ホストからの `ypset` コマンドをすべて拒否します。NIS サーバーが変更できないため、これは最も厳密に保護されたモードです。しかし、クライアント・マシンに直接接続されている NIS サーバーがネットワーク上に存在しない場合、`-ypsetme` フラグを使用する必要があり、`ypset` コマンドで NIS サーバーが指定されている必要があります。

ファイル

項目	説明
<code>/var/yp/binding</code>	ディレクトリー
<code>/var/yp/binding/domainname/ypservers</code>	NIS サーバーの IP アドレスおよびポート番号が入っています。ブロードキャストする前に連絡しようとする、サーバーのインターネット・アドレス (一行に 1 つずつ) のリストがあります。
<code>domainname.version</code>	現在の NIS サーバーのアドレスとポート番号が入っているバイナリー・ファイルです。

関連資料:

287 ページの『`ypset` コマンド』

290 ページの『`ypwhich` コマンド』

関連情報:

`domainname` コマンド

システム・リソース・コントローラー

ypcat コマンド

目的

Network Information Services (NIS) マップを出力します。

構文

Network Information Services データベースを表示する場合

```
/usr/bin/ypcat [ -k ] [ -t ] [ -d DomainName ] MapName
```

ニックネーム変換テーブルを表示する場合

```
/usr/bin/ypcat -x
```

説明

ypcat コマンドは、*MapName* パラメーターを使って指定した Network Information Services (NIS) マップを表示します。マップ名またはマップのニックネームを指定できます。**ypcat** コマンドは NIS サービスを使用するので、サーバーを指定する必要はありません。

フラグ

項目	説明
-k	キーの値が <code>null</code> のマップか、またはキーが値の一部でないマップ用のキーを表示します (<code>/etc</code> ディレクトリ内の ASCII バージョンを持つファイルから引き出されたマップは、いずれもこのクラスには属しません)。
-t	<i>MapName</i> パラメーターを使って指定された名前が、ニックネームではないことを示します。このフラグによって、 ypcat コマンドはニックネーム変換テーブルをバイパスし、 <i>MapName</i> パラメーターを使って指定されたマップだけを検索します。
-d <i>DomainName</i>	指定されたマップを指定されたドメイン内で検索します。
-x	ニックネーム変換テーブルを表示します。このテーブルはコマンドが認識しているマップ・ニックネームをリストし、各ニックネームに関連する (<i>MapName</i> パラメーターで指定されたとおりの) マップ名を示します。

例

1. ネットワーク全体のパスワード・マップ **passwd.byname** を見るには、下記のように入力します。

```
ypcat passwd
```

この例では、`passwd` は **passwd.byname** マップのニックネームです。

2. マップの位置を探すには、下記のように入力します。

```
ypcat -t passwd
```

この例では、**ypcat** コマンドは `passwd` というニックネームを持つマップをバイパスし、`passwd` という完全名を持つマップを検索します。

3. 他のドメインのマップを表示するには、下記のように入力します。

```
ypcat -d polaris passwd
```

この例では、**ypcat** コマンドは `passwd` という名前のマップを `polaris` という名前のドメインに入れます。

4. マップ・ニックネーム変換テーブルを表示するには、下記のように入力します。

```
ypcat -x
```

この例では、**ypcat** コマンドはマップ・ニックネームのリストと、それらに関連するマップ名を表示します。

関連情報:

domainname 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

ypinit コマンド

目的

Network Information Services (NIS) サーバー上で NIS マップをセットアップします。

構文

NIS マスター・サーバー上で NIS をセットアップする場合

```
/usr/sbin/ypinit [ -o ] [ -n ] [ -q ] -m [ SlaveName ... ]
```

NIS スレーブ・サーバー上で NIS をセットアップする場合

```
/usr/sbin/ypinit -s MasterName
```

説明

ypinit コマンドは、Network Information Services (NIS) マスター・サーバーまたは NIS スレーブ・サーバー上に NIS マップをセットアップします。 **ypinit** コマンドを使うことができるのは root ユーザー権限を持つユーザーのみです。

デフォルトでは **ypinit** コマンドは、作成されるマップの入力ファイルとして ASCII システム・ファイルを使います。

フラグ

項目	説明
-m [SlaveName...]	ローカル・ホストが NIS マスターになることを示します。 -q フラグが使われると、 -m フラグの後に NIS スレーブ・サーバーとなるコンピューターの名前を付けることができます。
-n	ypinit コマンドが、エラーを見つけても中断されないことを示します。
-o	現在の NIS ドメイン用の既存のマップをすべて上書きできるようにします。
-q	ypinit コマンドが、入力プロンプトの代わりにコマンド・ラインから引数を取得することを示します。
-s MasterName	MasterName パラメーター内で指定されたサーバー・ワークステーションから NIS マップをコピーします。

例

- すべての NIS マップのマスターとして機能する NIS マスター・サーバーをセットアップするには、コマンド・ラインから下記のコマンドを入力します。

```
ypinit -m
```

このコマンドは、**/var/yp/Makefile** ファイル内の命令に従う **make** プロシーチャーを呼び出します。

- NIS スレーブ・サーバーをセットアップするには、下記のように入力します。

```
ypinit -s zorro
```

この例で、**ypinit** コマンドは NIS マップを zorro という名前の NIS サーバーからユーザーのワークステーションへコピーし、ユーザーのワークステーションを NIS スレーブ・サーバーにします。

3. 入力プロンプトを使わずに NIS マスター・サーバーをセットアップするには、下記のように入力します。

```
ypinit -o -n -q -m slave
```

注: システムが既に NIS マスター・サーバーとして構成されている場合、**ypinit** を実行する前に、ディレクトリー **/var/yp/binding** が除去されていることを確認する必要があります。古い情報が **/var/yp/binding** に保管されていると、NIS マスター・サーバーの構成中にエラーが発生する原因になることがあります。

ファイル

項目	説明
/etc/bootparams	ディスクレス・クライアントがブートに使用できるクライアントをリストします。
/etc/passwd	コンピューターにログインするための許可を持つ各ユーザー用のエントリーが入っています。
/etc/group	コンピューターにログインすることが認められている各ユーザー・グループ用のエントリーが入っています。
/etc/hosts	ネットワーク上の各ホストのエントリーが入っています。
/var/yp/Makefile	NIS マップの作成規則が入っています。
/etc/networks	DARPA インターネット内の各ネットワークの名前が入っています。
/etc/netmasks	IP 標準サブネットのインプリメントに使用するネットワーク・マスクをリストします。
/etc/netid	コンピューター、ホスト、およびグループの識別情報が入っています。
/etc/rpc	RPC プログラム用のマップ情報が入っています。
/etc/services	インターネットを通して使用可能な各サーバー用のエントリーが入っています。
/etc/protocols	ローカル・ホスト上で使われるインターネット・プロトコルを定義します。
/etc/netgroup	ネットワーク上の各ユーザー・グループに関する情報が入っています。
/etc/ethers	インターネット・ネットワーク上のホストのイーサネット・アドレスが入っています。
/etc/publickey	NIS マップの公開鍵または秘密鍵が入っています。

関連情報:

システム管理のための NIS の概要

NIS リファレンス

ypmatch コマンド

目的

Network Information Services (NIS) マップ内の指定のキー値を表示します。

構文

NIS マップのキーの値を表示する場合

```
/usr/bin/ypmatch [ -d Domain ] [ -k ] [ -t ] Key... MapName
```

NIS マップ・ニックネーム・テーブルを表示する場合

```
/usr/bin/ypmatch -x
```

説明

ypmatch コマンドは、Network Information Services (NIS) マップ内の 1 つ以上のキーに関連する値を表示します。検索したいマップの名前またはニックネームのいずれかを指定するには *MapName* パラメーターを使用してください。

Key パラメーター内に複数のキーを指定すると、システムはすべてのキーを同じマップ内で検索します。パターン・マッチングが使用できないため、各キーの大文字と小文字の区別および長さを正確に合わせてください。指定したキーと一致するものが見つからなかった場合は、システムは診断メッセージを表示します。

フラグ

項目	説明
-d <i>Domain</i>	デフォルト・ドメイン以外のドメインを指定します。
-k	キーの値を表示する前に、キーの後にコロンを付けて表示します。これは、値の中でキーが重複しない場合、またはキーの指定が多すぎて出力が分かりにくくなるような場合に有効です。
-t	ニックネームをマップ名に変換することを禁止します。
-x	マップ・ニックネーム・テーブルを表示します。これによって、コマンドが認識している (<i>MapName</i> パラメーターで指定されたとおりの) ニックネームがリストされ、各ニックネームに関連するマップ名が示されます。

例

特定のキーに関連する値を表示するには、下記のように入力します。

```
ypmatch -d ibm -k host1 hosts
```

この例で **ypmatch** コマンドは、*ibm* ドメイン内の *hosts* マップから *host1* キーの値を表示します。

関連情報:

システム管理のための NIS の概要

NIS リファレンス

ypasswd コマンド

目的

Network Information Services (NIS) のネットワーク・パスワードを変更します。

構文

```
ypasswd [ -f [ Name ] | -s [ Name [ ShellProg ] ] ]
```

説明

ypasswd コマンドは、ネットワーク・パスワードを変更 (またはインストール) し、これを *UserName* パラメーター内で指定した名前に関連付けます。パスワードを作成または変更するには、変更したいパスワードのオーナーである必要があります。Network Information Services (NIS) パスワードは自分のコンピューターのパスワードと同じである必要はありません。NIS サーバー上の *root* ユーザーは、他のユーザーのパスワードを、元のパスワードを知らなくても変更することができます。変更する場合、この *root* ユーザーは他のユーザーの元のパスワードの代わりに、自分のパスワードを入力します。しかし、NIS クラウド上の *root* ユーザーには、この特権がありません。

コマンド・ラインに **ypasswd** コマンドを入力すると、システムは古いパスワードを入力するようプロンプトを表示します。これを行うと、システムは新しいパスワードを入力するようプロンプトを出します。入力するパスワードは、大文字と小文字を組み合わせた場合、4 文字まで短くできます。そうでなければ、パスワードは 6 文字以上の長さである必要があります。これらの規則は、順守すれば厳しいものではありません。

古いパスワードを誤って入力してしまったら、システムがエラー・メッセージを表示する前に新しいパスワードを入力する必要があります。システムは両方のパスワードを必要とします。これは、**update** プロトコルが両方のパスワードを同時にサーバーへ送るためです。サーバーはエラーを受け取り、古いパスワードが誤って入力されたことをユーザーに知らせます。

新しいパスワードを検査するために、システムはパスワードをもう一度入力するようプロンプトを表示します。この新しいパスワードを有効にするには、**yppasswdd** デーモンがユーザーの NIS サーバー上で稼働している必要があります。

注: **yppasswdd** コマンドでは、**passwd** コマンドが行うようなパスワードの規則を作成することはできません。

フラグ

項目	説明
-f [Name]	NIS マップ内でユーザー Name の gecos 情報を変更します。 gecos 情報は、 /etc/passwd ファイルに保管されている一般情報です。
-s [Name [ShellProg]]	NIS マップ内でユーザー Name のログイン・シェルを変更します。

例

1. ユーザーの NIS パスワードを変更するには、下記のように入力します。

```
yppasswd Joe
```

この例はユーザー名 Joe の NIS パスワードを変更する方法を表しています。システムは、Joe の古いパスワードを入力し、次に新しいパスワードを入力するよう指示するプロンプトを出します。

2. **yppasswdd** デーモンが **-noshell** フラグで始動されていない場合に、ユーザー Joe のログイン・シェルを **/bin/ksh** に変更するには、下記のように入力します。

```
yppasswd -s Joe /bin/ksh
```

3. **yppasswdd** デーモンが **-nogecos** フラグで始動されていない場合に、**passwd** ファイル内でユーザー Joe に関する **gecos** 情報を変更するには、下記のように入力します。

```
yppasswd -f Joe
Old NIS password:
Joe's current gecoc:
John Doe Test User Id
Change (yes) or (no)? >y
To?>Joe User Test User Id
```

関連情報:

システム管理のための NIS の概要

NIS リファレンス

『yppasswdd デーモン』

yppasswdd デーモン

目的

yppasswd コマンドからの要求を受け取って実行します。

構文

```
rpc.yppasswdd FileName [ -nogecos ] [ -nopw ] [ -noshell ] [ -r | -m [ Argument... ] ]
```

説明

yppasswdd デーモンは、**yppasswd** コマンドからの新しいパスワード要求を受け取って実行するサーバーです。これらの要求は、ユーザーの古いパスワードを検査して変更することをデーモンに求めるものです。デーモンは *FileName* パラメーター内で指定したファイル内のパスワードを変更します。このファイルは **/etc/passwd** ファイルと同じフォーマットです。

リモート・マシンから Network Information Services (NIS) パスワード・マップの更新ができるようにするには、**yppasswdd** デーモンが、NIS パスワード・マップを含むマスター・サーバー上で稼働している必要があります。

注: **yppasswdd** デーモンは、デフォルトでは実行されません。また、他のリモート・プロシージャ・コール (RPC) デーモンのように、**inetd** デーモンから開始することはできません。

yppasswdd デーモンは、次のシステム・リソース・コントローラー (SRC) コマンドによって始動、停止されます。

```
startsrc -s yppasswdd
```

```
stopsrc -s yppasswdd
```

フラグ

項目	説明
-m	/var/yp ディレクトリー内の Make ファイルを使って make コマンドを実行します。これによって NIS パスワード・マップに新しいパスワード、または変更されたパスワードが追加されます。 -m フラグの後の引数はいずれも make コマンドに渡されます。
-nogecos	サーバーが yppasswd コマンドからの gecos 情報の変更を受け入れないことを示します。
-nopw	サーバーが yppasswdd コマンドからのパスワード変更を受け入れないことを示します。
-noshell	サーバーが yppasswd コマンドからのユーザー・シェル変更を受け入れないことを示します。
-r	新規もしくは変更されたパスワードで、Master サーバーと Slave サーバー上の /var/yp/domainname/passwd.byname および /var/yp/domainname/passwd.byuid データベース・ファイルを直接更新します。 make コマンドが実行されないので、このオプションは、 -m フラグよりも高速です。 -r フラグは、データベース・ファイルが (エントリーが数千を超えるくらい) 大きいときに便利です。

注: システム・リソース・コントローラー (SRC) は、デフォルトで指定される **-m** フラグ付きの **yppasswdd** デーモンを開始します。**chssys** コマンドを使用して、デフォルトを **-r** フラグに変更します。

例

更新されたパスワードを即座に反映させるには、下記のように **yppasswdd** デーモンを始動します。

```
startsrc -s yppasswdd
```

ファイル

項目	説明
<code>/etc/inetd.conf</code>	inetd デーモンがインターネット・サービス要求をどのように処理するかを定義します。
<code>/var/yp/Makefile</code>	NIS マップの作成規則が入っています。
<code>/etc/rc.nfs</code>	NFS デーモンと NIS デーモンの始動スクリプトが入っています。
<code>/etc/security/passwd</code>	パスワード情報を保管します。

関連資料:

281 ページの『`yppasswd` コマンド』

関連情報:

`domainname` コマンド

`passwd` コマンド

`inetd` daemon

`f/etc/security/passwd` ファイル

システム・リソース・コントローラー

yppoll コマンド

目的

現在サーバーで使用中の Network Information Services (NIS) マップのオーダー番号 (ID 番号) を表示します。

構文

```
/usr/sbin/yppoll [ -h Host ] [ -d Domain ] MapName
```

説明

yppoll コマンドは、**ypserv** デーモンを使って、*MapName* パラメーター内で指定したマップのオーダー番号を表示します。オーダー番号はマップの ID 番号で、システムによって割り当てられます。この番号はマップが更新されるたびに変わります。特定のマップの最新バージョンをサーバーが使用していることを確認したいときはいつでも、**yppoll** コマンドを使ってください。

yppoll コマンドは、Network Information Services (NIS) プロトコルのバージョン 1 またはバージョン 2 がインストールされているシステム上で実行できます。ただし、プロトコルの各バージョンにはそれぞれ独自の診断メッセージ・セットがあることに注意してください。

注: *MapName* を指定する際、マップの完全名が入力されていることを確認してください。**yppoll** コマンドはマップのニックネームを認識しません。

フラグ

項目	説明
-h <i>Host</i>	デフォルト・サーバー以外のサーバーを指定できるようにします。コマンドのデフォルト・サーバーを知りたいときは、 ypwhich コマンドを使ってください。
-d <i>Domain</i>	デフォルト・ドメイン以外のドメインを指定できるようにします。コマンドのデフォルト・ドメインを知りたいときは、 domainname コマンドを使ってください。

例

1. 特定のホストに置かれているマップを見るには、下記のように入力します。

```
/usr/sbin/ypoll -h thor netgroups.byuser
```

この例で **ypoll** コマンドは、**thor** という名前のホスト上に置かれている **netgroups.byuser** マップのオーダー番号を表示します。

2. ドメイン上のマップを見るには、下記のように入力します。

```
/usr/sbin/ypoll -d atlantis hosts.byname
```

この例で **ypoll** コマンドは、ドメイン **atlantis** 上に置かれている **hosts.byname** マップのオーダー番号を表示します。

関連資料:

290 ページの『**ypwhich** コマンド』

関連情報:

286 ページの『**ypserv** デーモン』

domainname コマンド

yppush コマンド

目的

Network Information Services (NIS) のスレーブ・サーバーに、更新済みの NIS マップのコピーを指示するプロンプトを出します。

構文

```
/usr/sbin/yppush [ -v ] [ -d Domain ] MapName
```

説明

yppush コマンドは、**/usr/etc/yp** ディレクトリーから発行されるもので、更新された Network Information Services (NIS) マップをコピーするために NIS スレーブ・サーバーにプロンプトを出します。 **MapName** 変数を使って、スレーブ・サーバーへ転送されるマスター・サーバーのマップを指定します。プロンプトを必要とするサーバーのリストを得るために、**yppush** コマンドは、**Domain** パラメーターまたは現在のデフォルト・ドメインによって指定された **ypservers** マップを読み取ります。プロンプトが出されると、各スレーブ・サーバーは **ypxfr** コマンドを使って、マップをコピーし、自分のデータベースへ返送します。

System Management Interface Tool (SMIT) を使用して、このコマンドを実行できます。SMIT を使用するには、次のように入力します。

```
smit yppush
```

注: NIS プロトコルのバージョン 1 を使っているシステムでは、**ypxfr** コマンドは転送エージェントではありません。

フラグ

項目	説明
-d <i>Domain</i>	デフォルト・ドメイン以外のドメインを指定します。指定したドメインのマップは、存在している必要があります。
-v	各サーバーが呼び出されるたびにメッセージを表示し、その次に、バージョン 2 プロトコルを使っている場合は、各サーバーの応答ごとにメッセージを 1 つ表示します。このフラグが省略されると、コマンドはエラー・メッセージだけを表示します。 注: バージョン 1 の NIS プロトコルではメッセージを表示しません。バージョン 1 を使用するシステムでは、 yppoll コマンドを使って転送が行われたことを検査してください。

例

1. 他のドメインからスレーブ・サーバーへマップをコピーするには、下記のように入力します。

```
/usr/sbin/yppush -d atlantis netgroup
```

この例で **yppush** コマンドは、atlantis ドメインから netgroup マップをコピーします。

2. 各スレーブ・サーバーを呼び出すたびに **yppush** コマンドの進捗状況を表示するには、下記のように入力します。

```
/usr/sbin/yppush -v -d atlantis netgroup
```

この例で **yppush** コマンドは、atlantis ドメインから各ネットワークのスレーブ・サーバーに netgroup マップをコピーするたびに、進捗状況についてのメッセージを表示します。

ファイル

項目	説明
<code>/var/yp/DomainName/ypservers.{dir, pag}</code>	更新された NIS マップのコピーを作成するために yppush コマンドがプロンプトを出すサーバーのリストです。

関連資料:

284 ページの『**yppoll** コマンド』

291 ページの『**ypxfr** コマンド』

関連情報:

System Management Interface Tool (SMIT)

ypserv デーモン

目的

ローカルの Network Information Services (NIS) マップ内の情報を検索します。

構文

```
/usr/lib/netsvc/yp/ypserv
```

説明

ypserv デーモンは、Network Information Services (NIS) マップ内で情報を検索します。 **ypserv** デーモンによって実行される操作は、実施責任者には NIS プロトコル仕様によって、またプログラマーには `/usr/include/rpcsvc/yp_prot.h` ヘッダー・ファイルによって定義されます。 **ypserv** デーモンとの通信はリモート・プロシージャ・コール (RPC) を通して行われます。

ypserv デーモンは、サーバー・マシン上でのみ稼働します。 **ypserv** デーモンは、次のシステム・リソース・コントローラー (SRC) コマンドによって始動、停止されます。

```
startsrc -s ypserv
```

```
stopsrc -s ypserv
```

ypserv デーモンは、NIS ドメイン内の指定されたマップ上で下記の操作を実行します。

項目	説明
Match	キーを受け取り、対応する値を返します。
Get_first	マップから得た最初のキー値の組を返します。
Get_next	マップ内の次のキー値の組を列挙します。
Get_all	単一の RPC 要求へ応答して、完全 NIS マップを要求者へ送ります。
Get_order_number	マップ・エントリーの代わりにマップ情報を提供します。オーダー番号はキー値の対として実際にマップ内にありますが、サーバーはこの番号を、通常のルックアップ関数からは返しません。ただし、 makedbm コマンドを使ってマップを調べればこの対を見ることができます。
Get_master_name	マップ・エントリーの代わりにマップ情報を提供します。マスター名はキー値の対として実際にマップ内にありますが、サーバーはこの名前を、通常のルックアップ関数からは返しません。ただし、 makedbm コマンドを使ってマップを調べればこの対を見ることができます。

ログ情報は、**ypserv** デーモンの実行開始時に `/var/yp/ypserv.log` ファイルがあれば、それに書き込まれます。

`/var/yp/securenets` ファイルが存在する場合、**ypservr** コマンドはこのファイル内で指定された ip 範囲内のホストにのみ応答します。

ファイル

項目	説明
<code>/etc/rc.nfs</code>	NFS デーモンと NIS デーモンの始動スクリプトが入っています。
<code>/var/yp/ypserv.log</code>	ypserv デーモンのログが入っています。

関連情報:

makedbm コマンド

Remote Procedure Call Overview for Programming

システム・リソース・コントローラー

ypset コマンド

目的

クライアント・マシンを特定のサーバーに割り当てます。

構文

```
/usr/sbin/ypset [ -V1 ] [ -d Domain ] [ -h Host ] Server
```

説明

ypset コマンドは、クライアント上の **ypbind** デーモンをサーバーの **ypserv** デーモンへ割り当てます。**ypbind** デーモンは、*Domain* パラメーター内で指定したドメインのための Network Information Services (NIS) のサービスを得るために、*Server* パラメーター内で指定したサーバーに対して作動します。**ypbind** デーモンは、サーバーの **ypserv** デーモンから NIS サービスを得ます。

割り当てが設定されると、クライアント・プロセス (**ypcat** コマンドや **ypwhich** コマンドなど) によってドメインのための割り当てを得ようとするまで、検査は行われません。バインドが失敗に終わると (指定されたサーバーがダウンしたり、**ypserv** デーモンを実行しない場合)、再び **ypbind** デーモンは同じドメインのためにバインドを試みます。

Server パラメーターには名前またはインターネット・プロトコル (IP) のアドレスのどちらかを指定してください。名前を指定した場合、**ypset** コマンドは、NIS サービスを使用し IP アドレスに対して名前を解決しようとしています。これは、コンピューター上で、問題となっているドメインのための有効な割り当てが現在行われているときのみ働く機能です。ほとんどの場合、サーバーを IP アドレスとして指定することになります。

ローカル・ネットワーク上のいくつかのホストが NIS サービスを提供している場合、**ypbind** デーモンは他のホストへ再バインドすることができます。サーバーがダウンしたり **ypserv** デーモンを実行していないとき、**ypbind** デーモンはクライアントを他のサーバーへ再バインドします。このようにして Network Information Service は、使用可能な NIS サーバー間で負荷のバランスを保ちます。

下記の場合には **ypset** コマンドを使います。

- ネットワークがブロードキャストをサポートしていないとき。
- ネットワークがブロードキャストをサポートしてはいるが、NIS サーバーを持っていないとき。
- ネットワークが特定の NIS サーバーだけにあるマップにアクセスするとき。

ypset を使用する代替方法は、*/var/yp/binding/domain_name/ypservers* ファイルを使用することです。このファイル (存在する場合) は、1 回線につき 1 台のサーバーをバインドしようとする NIS サーバーのリストを含んでいるはずです。**ypbind** デーモンが **ypservers** ファイルにあるどのサーバーにもバインドできない場合は、**ypset** によって指定されているサーバーを使用しようとしています。失敗する場合は、サブネット上で NIS サーバーを求めてブロードキャストします。

フラグ

項目	説明
-d <i>Domain</i>	デフォルト・ドメイン以外のドメインを指定します。
-h <i>Host</i>	ローカル・ホストの代わりに指定されたホスト上に、 ypbind デーモンのための割り当てを設定します。ホストは名前として、または IP アドレスとして指定できます。
-V1	(古い) バージョン 1 の NIS プロトコル用に指定されたサーバーをバインドします。

例

特定のドメイン内のホスト上でバインドされるサーバーを設定するには、下記のように入力します。

```
ypset -d ibm -h venus mars
```

この例では、**ypset** コマンドによって、ホスト **venus** がサーバー **mars** へバインドされます。

関連資料:

277 ページの『**ypcat** コマンド』

290 ページの『**ypwhich** コマンド』

関連情報:

276 ページの『**ypbind** デーモン』

286 ページの『**ypserv** デーモン』

ypupdated デーモン

目的

Network Information Services (NIS) マップ内の情報を更新します。

構文

```
/usr/lib/netsvc/yp/rpc.yupdated [ -i | -s ]
```

説明

ypupdated デーモンは、Network Information Services (NIS) マップ内の情報を更新します。ただし、情報の更新を行う前に、デーモンは **/var/yp** ディレクトリー内の **updaters** ファイルを調べ、どの NIS マップを更新するべきか、またどのように変更すべきかを判断します。

デフォルトでは **ypupdated** デーモンは、それにとって有効で最も安全な認証方法である DES (保護されている) または UNIX (保護されていない) のどちらかを要求します。

ypupdated デーモンは、下記のシステム・リソース・コントローラー (SRC) コマンドによって始動または停止されます。

```
startsrc -s ypupdated
```

```
stopsrc -s ypupdated
```

フラグ

項目	説明
-s	保護されたリモート・プロシージャ・コール (RPC) 機構 (AUTH_DES 認証) を使って認証された呼び出しだけを受け付けます。これによって、ネットワークがこれらの呼び出しをサポートしていないときには、NIS マップはプログラムを用いて更新できなくなります。
-i	保護されていない AUTH_UNIX 証明書を使って RPC コールを受け付けます。これによって、すべてのネットワーク内の NIS マップはプログラムを用いて更新することができるようになります。

例

コマンド・ラインから **ypupdated** デーモンを始動するには、下記のように入力します。

```
startsrc -s ypupdated
```

ファイル

項目	説明
<code>/var/yp/updaters</code>	NIS マップを更新するための Make ファイル。

関連情報:

startsrc コマンド

Remote Procedure Call Overview for Programming

システム・リソース・コントローラー

ypwhich コマンド

目的

Network Information Services (NIS) サーバーか、指定のマップのマスターであるサーバーのどちらであるかを識別します。

構文

NIS サーバーを識別する場合

```
/usr/bin/ypwhich [ -d Domain ] [ -V1 | -V2 ] [ HostName ]
```

マップ用のマスター NIS サーバーを識別する場合

```
/usr/bin/ypwhich [ -t ] [ -d Domain ] [ -m [ MapName ] ]
```

マップ・ニックネーム・テーブルを表示する場合

```
/usr/bin/ypwhich -x
```

説明

ypwhich コマンドは、**ypwhich** コマンドの呼び出し方法に応じて、どちらのサーバーが Network Information Services (NIS) のサービスを提供するか、あるいはどちらのサーバーがマップのマスターであるかを識別します。引数を付けずに呼び出したときは、このコマンドはローカル・マシン用の NIS サーバーの名前を表示します。ホスト名を指定すると、システムはどちらのマスターが使われているかを見付けるために、そのホストを照会します。

フラグ

項目	説明
<code>-d Domain</code>	デフォルト・ドメインの代わりに指定されたドメインを使います。
<code>-V1</code>	どちらのサーバーが、古いバージョン 1 の NIS プロトコル・クライアント・プロセスを行うかを示します。
<code>-V2</code>	どちらのサーバーが、現行バージョン 2 の NIS プロトコル・クライアント・プロセスを行うかを示します。どちらのバージョンも指定されないと、 ypwhich コマンドは、バージョン 2 のサービスを提供するサーバーをマップしようとします。現在バインドされたバージョン 2 のサーバーがなければ、次に ypwhich コマンドはバージョン 1 のサービスを提供しているサーバーをマップしようとします。サーバーおよびクライアントはどちらも前バージョンに対しては互換性があるため、ユーザーはどちらのバージョンが現在使われているかを気にする必要はほとんどありません。
<code>-t</code>	マップ名と同一のニックネームがあるときに便利なニックネーム変換を禁止します。

項目	説明
-m <i>MapName</i>	マップのためのマスター NIS サーバーを検索します。ホストは -m フラグを付けて指定できません。 <i>MapName</i> 変数は、マップ名またはマップのニックネームのどちらでもかまいません。マップ名が省略されると、 -m フラグは使用可能なマップのリストを作成します。
-x	マップ・ニックネーム・テーブルを表示します。これは、コマンドが認識しているニックネーム (<i>MapName</i>) をリストし、各ニックネームに対応するマップ名を示します。

例

1. マップのマスター・サーバーを見付けるには、下記のように入力します。

```
ypwhich -m passwd
```

この例で **ypwhich** コマンドは、passwd マップのサーバーの名前を表示します。

2. passwd のニックネームを持つマップではなく、passwd という名前のマップを見付けるには、下記のように入力します。

```
ypwhich -t -m passwd
```

この例で **ypwhich** コマンドは、完全名が passwd であるマップのサーバー名を表示します。

3. 古いバージョン 1 の NIS プロトコルを実行するクライアントにリソースを提供するサーバーを表示するには、下記のように入力します。

```
ypwhich -v1
```

4. マップ・ニックネーム・テーブルを表示するには、下記のように入力します。

```
ypwhich -x
```

関連情報:

システム管理のための NIS の概要

NIS リファレンス

ypxfr コマンド

目的

Network Information Services (NIS) マップを、NIS サーバーからローカル・ホストへ転送します。

構文

```
/usr/sbin/ypxfr [ -f ] [ -c ] [ -d Domain ] [ -h Host ] [ -s Domain ] [ -C TID Program IPAddress Port ] [ -S ] MapName
```

説明

ypxfr コマンドは、下記の手順で Network Information Services (NIS) サーバーからローカル・ホストへ NIS マップを転送します。

1. クライアントの */var/yp/Domain* ディレクトリー (既存のものでなければなりません) 内に一時マップを作成します。
2. サーバーからマップ・エントリーを取り出し、クライアント上のマップを 1 つずつ充てんしていきます。
3. マップ・パラメーター (オーダー番号とサーバー) を取り出し、ロードします。
4. マップの古いバージョンをすべて削除します。

5. 一時マップを実マップ名に変更します。

`/var/yp/securenets` ファイルが存在する場合、`ypxfr` コマンドはこのファイルにリストされているホストにのみ応答します。

`MapName` 変数は、NIS サーバーから転送されるマップの名前を指定します。

対話式に実行するときは、`ypxfr` コマンドは出力をユーザーの端末へ送ります。制御端末なしに呼び出した場合、(ファイルが既に存在していれば) `ypxfr` コマンドはその出力を `/var/yp/ypxfr.log` ファイルに追加します。このファイルは、転送の試みとその結果を逐一記録します。`ypxfr` コマンドは、`root` ユーザーの `crontab` ファイル、または `yppserv` デーモンから呼び出される場合がほとんどです。

サーバー間の情報を常に一致させておくために、`ypxfr` コマンドを使って NIS データベース内のすべてのマップを定期的に更新してください。ただし、マップ内には他のマップよりも変更が多く、そのため頻繁に更新しなければならないものがあることに注意してください。例えば、数カ月ごとにしか変更されないような変更頻度の低いマップでも、少なくとも月に一度は更新してください。1 日に何度も変更されるような変更頻度の高いマップは、1 時間ごとに更新検査しなければなりません。例えば `services.byname` マップは数カ月間は変更されないかもしれませんが、一方で `hosts.byname` マップは、1 日に何度も変更されることがあります。

定期的更新を自動的に行うには、`crontab` エントリーを使ってください。一回でいくつものマップを更新するには、シェル・スクリプト内にコマンドをグループ化しておきます。シェル・スクリプトのサンプルは、`/usr/etc/yp` ディレクトリー内の `ypxfr_1perday`、`ypxfr_2perday`、`ypxfr_1perhour` ファイルに入っています。

System Management Interface Tool (SMIT) を使用して、このコマンドを実行できます。SMIT を使用するには、次のように入力します。

```
smit ypxfr
```

フラグ

項目	説明
<code>-C TID Program IPAddress Port</code>	<code>ypxfr</code> コマンドに、 <code>yppush</code> コマンドを探す場所を指示します。 <code>yppserv</code> デーモンは、 <code>ypxfr</code> コマンドを呼び出し、 <code>yppush</code> コマンドをホストにコールバックします。指定には下記のパラメーターを使用してください。 <code>TID</code> <code>yppush</code> コマンドのトランザクション ID を指定します。 <code>Program</code> <code>yppush</code> コマンドに関連付けられたプログラム番号を指定します。 <code>IPAddress</code> <code>yppush</code> コマンドが存在しているポートのインターネット・プロトコル・アドレスを指定します。 <code>Port</code> <code>yppush</code> コマンドが <code>listen</code> しているポートを指定します。 注: このオプションは、 <code>yppserv</code> デーモン専用です。
<code>-c</code>	Clear Current Map への要求がローカルの <code>yppserv</code> デーモンへ送信されないようにします。 <code>ypxfr</code> コマンドを実行しているときに <code>yppserv</code> デーモンがローカルで実行されていなければ、このフラグを使ってください。そうでなければ、 <code>ypxfr</code> コマンドはエラー・メッセージを表示して、転送は失敗します。
<code>-d Domain</code>	デフォルト・ドメイン以外のドメインを指定します。指定したドメインのマップは、存在している必要があります。
<code>-f</code>	マスターでのバージョンがローカルのバージョンより古くても、強制的に転送を実行します。

項目	説明
-h <i>Host</i>	マスターが何であるかについてマップが示した内容にかかわらず、指定されたホストからマップを取り出します。ホストが指定されないと、 ypxfr コマンドは、NIS サービスにマスターの名前を要求し、そこからマップを取り出そうと試みます。 <i>Host</i> 変数には a.b.c.d. のフォーマットで名前または IP アドレスを指定することができます。
-S	転送するマップの取得元の ypserv サーバーに対して、特権 IP ポートを使用するよう要求します。特権ポートの使用が認められているのは一般的に root ユーザー処理だけであるため、この機能によって、転送に対して特別なセキュリティー基準が追加されます。転送されるマップが安全なマップならば、 ypxfr コマンドはマップ上の許可を 0600 に設定します。
-s <i>Domain</i>	マップの転送元のソース・ドメインを指定します。マップはドメイン間で同一のものでなければなりません (例えば services.byname マップなど)。

例

他のドメイン内のホストからマップを取り出すには、下記のように入力します。

```
/usr/sbin/ypxfr -d ibm -h venus passwd.byname
```

この例で **ypxfr** コマンドは、ibm ドメイン内のホスト venus から passwd.byname マップを取り出します。

ファイル

項目	説明
/var/yp/ypxfr.log	ログ・ファイルが入っています。
/usr/sbin/ypxfr_1perday	cron デーモンと一緒に使用して、1 日に 1 回ずつ転送するためのスクリプトが入っています。
/usr/sbin/ypxfr_2perday	1 日に 2 回ずつ転送するためのスクリプトが入っています。
/usr/sbin/ypxfr_1perhour	揮発性マップを 1 時間ごとに転送するためのスクリプトが入っています。

関連資料:

285 ページの『yppush コマンド』

関連情報:

System Management Interface Tool (SMIT)

crontab コマンド

cron コマンド

286 ページの『ypserv デーモン』

Z

以下の AIX コマンドは、文字 zで始まります。

zcat コマンド

目的

圧縮ファイルを標準出力に展開します。

構文

```
zcat [ -n ] [ -V ] [ File ... ]
```

説明

zcat コマンドにより、ユーザーは圧縮ファイルを圧縮解除せずに展開して表示することができます。**zcat** コマンドでは、展開したファイルの名前の変更を行ったり、**.Z** 拡張子を除去したりはしません。**zcat** コマンドは、展開した出力を標準出力に書き出します。

フラグ

項目	説明
-n	圧縮ファイルから圧縮ファイル・ヘッダーを省略します。 注: ファイルが -n オプションを使用して圧縮されてある場合は、 -n オプションを使用します。
-V	現行バージョンとコンパイル・オプションを標準エラーに書き込みます。

パラメーター

項目	説明
<i>File ...</i>	展開する圧縮ファイルを指定します。

戻り値

zcat コマンドが状況 1 で終了した場合、下記のいずれかのイベントが発生している可能性があります。

- 入力ファイルが **compress** コマンドで作成されていなかった。
- 入力ファイルを読めなかった、または出力ファイルを書き込めなかった。

エラーが発生しなければ、終了状況は 0 です。

終了状況

項目	説明
0	正常終了。
>0	エラーが発生しました。

例

foo.Z ファイルを圧縮解除せずに表示するには、下記のように入力します。

```
zcat foo.Z
```

foo.Z ファイルの圧縮解除された内容が標準出力に書き出されます。ファイルの名前の変更は行われません。

関連情報:

compress コマンド

zdump コマンド

目的

タイムゾーン情報を表示します。

構文

```
zdump [--version] [--help] [-v] [-V] [-c [loyear,]hiyear] [-t [lotime,]hitime] zonename ...
```

説明

zdump コマンドは、コマンド・ラインで指定したゾーン名ごとに、タイムゾーン情報を出力します。

フラグ

項目	説明
--version	zdump コマンドのバージョン情報を表示します。
-v	コマンド・ラインで指定された各ゾーン名について、 zdump コマンドは、次の項目を出力します。 <ul style="list-style-type: none"> 考えられる最小の時間値。 考えられる最小の時間値の 1 日後の時間。 検出された不連続時間の 1 秒前と、正確なその時間の両方。 考えられる最大の時間値。 考えられる最大の時間値の 1 日前の時間。
	各行の後に、 <code>isdst=D</code> があります。 <i>D</i> は、指定された時間が夏時間調整、標準時間、または不明な時間タイプのいずれであるかに応じて、正の値、ゼロ値、または負の値です。各行の後には、 <code>gmtoff=N</code> もあり、ここで、指定されたローカル時間は、東半球で <i>N</i> 秒です。
-V	-v フラグが極限時間値からの相対値である時間を表示しないことを除き、このフラグの出力に似た出力を表示します。このフラグは、別の時間表現と比較しやすい出力を生成します。
-c [loyear,]hiyear	指定された年の詳細出力を報告します。カットオフ時間は、うるう秒を無視する年 0 および世界時 (UT) と一緒に先発グレゴリオ暦を使用して計算されます。下位年属性 (<i>loyear</i>) は排他的であり、上位年属性 (<i>hiyear</i>) は包含的です。例えば、1970 という <i>loyear</i> 値は、1970 年 1 月 1 日の 00:00:00 協定世界時に発生した遷移を除外しますが、1970 という <i>hiyear</i> 値はその遷移を含みます。デフォルトのカットオフ時間は -500,2500 です。
-t [lotime,]hitime	yyyy-mm-dd hh:mm:ss 協定世界時 (UTC) の構文の <i>lotime</i> 属性と <i>hitime</i> 属性に指定された特定時刻の詳細出力を報告します。例えば、1970-01-01 00:00:00 です。カットオフ時間には、 <i>zonename</i> パラメーターに応じてうるう秒が含まれます。 <p>-c フラグと同様、下位時間制限は排他的であり、上位時間制限は包含的です。</p>

パラメーター

項目	説明
<i>zonename</i>	タイムゾーン情報が表示されるゾーン名を表します。

終了状況

0 コマンドは正常に完了しました。

>0 エラーが発生しました。

例

1. シンガポールのタイムゾーン情報をレポートするには、次のコマンドを入力します。

```
zdump -v Singapore
```

2. 2035 年の初め頃にトルコの停止に関するタイムゾーン情報を表示するには、次のコマンドを入力します。

```
zdump -v -c 2035 Turkey
```

3. 2015 年から 2017 年までのニューヨークのタイムゾーン情報を報告するには、次のコマンドを入力します。

```
zdump -v -c 2015,2017 America/New_York
```

出力は、次の例のようになります。

```
America/New_York Fri Dec 13 20:45:52 1901 UT = Fri Dec 13 15:45:52 1901 EST isdst=0 gmtoff=-18000
America/New_York Sat Dec 14 20:45:52 1901 UT = Sat Dec 14 15:45:52 1901 EST isdst=0 gmtoff=-18000
America/New_York Sun Mar 8 06:59:59 2015 UT = Sun Mar 8 01:59:59 2015 EST isdst=0 gmtoff=-18000
America/New_York Sun Mar 8 07:00:00 2015 UT = Sun Mar 8 03:00:00 2015 EDT isdst=1 gmtoff=-14400
America/New_York Sun Nov 1 05:59:59 2015 UT = Sun Nov 1 01:59:59 2015 EDT isdst=1 gmtoff=-14400
America/New_York Sun Nov 1 06:00:00 2015 UT = Sun Nov 1 01:00:00 2015 EST isdst=0 gmtoff=-18000
America/New_York Sun Mar 13 06:59:59 2016 UT = Sun Mar 13 01:59:59 2016 EST isdst=0 gmtoff=-18000
America/New_York Sun Mar 13 07:00:00 2016 UT = Sun Mar 13 03:00:00 2016 EDT isdst=1 gmtoff=-14400
America/New_York Sun Nov 6 05:59:59 2016 UT = Sun Nov 6 01:59:59 2016 EDT isdst=1 gmtoff=-14400
America/New_York Sun Nov 6 06:00:00 2016 UT = Sun Nov 6 01:00:00 2016 EST isdst=0 gmtoff=-18000
America/New_York Mon Jan 18 03:14:07 2038 UT = Sun Jan 17 22:14:07 2038 EST isdst=0 gmtoff=-18000
America/New_York Tue Jan 19 03:14:07 2038 UT = Mon Jan 18 22:14:07 2038 EST isdst=0 gmtoff=-18000
```

4. 下位時間値および上位時間時なしで 2015 年から 2017 年までのニューヨークのタイムゾーン情報を報告するには、次のコマンドを入力します。

```
zdump -V -c 2015,2017 America/New_York
```

出力は、次の例のようになります。

```
America/New_York Sun Mar 8 06:59:59 2015 UT = Sun Mar 8 01:59:59 2015 EST isdst=0 gmtoff=-18000
America/New_York Sun Mar 8 07:00:00 2015 UT = Sun Mar 8 03:00:00 2015 EDT isdst=1 gmtoff=-14400
America/New_York Sun Nov 1 05:59:59 2015 UT = Sun Nov 1 01:59:59 2015 EDT isdst=1 gmtoff=-14400
America/New_York Sun Nov 1 06:00:00 2015 UT = Sun Nov 1 01:00:00 2015 EST isdst=0 gmtoff=-18000
America/New_York Sun Mar 13 06:59:59 2016 UT = Sun Mar 13 01:59:59 2016 EST isdst=0 gmtoff=-18000
America/New_York Sun Mar 13 07:00:00 2016 UT = Sun Mar 13 03:00:00 2016 EDT isdst=1 gmtoff=-14400
America/New_York Sun Nov 6 05:59:59 2016 UT = Sun Nov 6 01:59:59 2016 EDT isdst=1 gmtoff=-14400
America/New_York Sun Nov 6 06:00:00 2016 UT = Sun Nov 6 01:00:00 2016 EST isdst=0 gmtoff=-18000
```

ファイル

項目
/usr/sbin/zdump
/usr/share/lib/zoneinfo

説明
SystemV **zdump** コマンドが入っています。
標準のタイムゾーン・ディレクトリーが入っています。

zic コマンド

目的

タイムゾーン・コンパイラーを提供します。

構文

```
zic [ -v ] [ -l LocalTime ] [ -p Posixrules ] [ -d Directory ] [ -L Leapseconds ] [ -y YearIsType ] [
FileName ...]
```

説明

zic コマンドは、コマンド・ラインで名前を付けたファイルのテキストを処理し、この入力で指定した時刻変換バイナリー・ファイルを作成します。ファイル名として - (ハイフン) を指定すると、標準入力からタイムゾーン情報が処理されます。

指定ファイル内の入力行は、フィールドから構成されています。フィールド区切り文字は、空白文字 (個数は関係ない) です。入力ファイル内のポンド記号 (#) は、コメントを表し、ポンド記号が現れる行の最後まで続きます。フィールドの一部として、空白文字やポンド記号を使用する場合は、二重引用符 (") でそれらを囲むことができます。ブランク行は無視されます。

ルール行は次のような形式です。

```
Rule NAME FROM TO TYPE IN ON AT SAVE LETTER/S
```

ルール行を構成するフィールドは下記のとおりです。

NAME

このルールが適用できる、ルールのセットにランダムな名前を示します。

FROM

ルールを適用する最初の年を示します。minimum (min) を指定すると、最小の年を表示可能な時刻値で表します。maximum (max) を指定すると、最大の年を表示可能な時刻値で表します。

TO

ルールを適用する最後の年を示します。これは、有効な時刻値として、または minimum や maximum を指定して、表示されます。only を指定すると、「FROM」フィールドの値を繰り返すことができます。

TYPE

ルールを適用する年のタイプを指定します。

TYPE フィールドには、下記の値が入ります。

'-' 「FROM」から「TO」までの (両端を含めた) すべての年に、ルールを適用します。

uspres

米国大統領選挙の年に、ルールを適用します。

nonpres

米国大統領選挙以外の年に、ルールを適用します。

上記以外の値が「**TYPE**」フィールドに入ると、**zic** コマンドは、`/usr/sbin/yearistype year type` コマンドを実行し、年の型を検査します。

yearistype コマンドには 2 つのパラメーター (**year** と **type of year**) が使用できます。終了状況 0 は、その年が指定されたタイプであることを意味します。そうでなければ、終了状況 1 を戻します。

IN ルールの適用が有効な月を表します。月の名前は、省略形でも可能です。

ON ルールの適用が有効な日を表します。認識される形式には、下記のものがあります。

- **lastFri** は、その月の最後の金曜日を表します。
- **lastMon** は、その月の最後の月曜日を表します。
- 1 カ月のうちの日付を表す数字。例えば、5 はその月の 5 番目の日を表します。
- **lastSun** は、その月の最後の日曜日を表します。
- **lastMon** は、その月の最後の月曜日を表します。
- **Sun>=8** は、8 日およびそれ以降の最初の日曜日を表します。
- **Sun<=25** は、25 日およびそれ以前の最後の日曜日を表します。

曜日の名前は、省略形でも、フルスペルでも可能です。「**ON**」フィールド内にはスペースが無いように注意してください。

AT ルールの適用が有効となる時刻を表します。認識される形式には、下記のものがあります。

- 時間を表す数字。例えば、2 は 2 時間を表します。
- 2:00 は、時と分の形式で 2 時を表します。
- 15:00 は、24 時間形式で、午後の 3 時を表します。
- 1:28:14 は、時分秒の形式で、1 時 28 分 14 秒を表します。

これらのいずれの形式でも、指定した時刻がローカル 壁掛け時計 (*wall-clock*) 時間である場合は、その後ろに **w** の文字が続き、また、指定した時刻がローカル標準 時間である場合は、**s** の文字が続きます。**w** または **s** が無い場合は、壁掛け時計 (*wall-clock*) 時間と見なします。

3 つ以上のローカル時間がある領域では、最も早い遷移時間ルールの「**AT**」フィールドでローカル標準時間を用いてください。それによってタイムゾーンのバイナリー結果として保存される最初の遷移時間が確かに正確なものとなります。

SAVE ルールが有効な場合、ローカル標準時間に追加される時間量を事前に指定します。このフィールドは、「**AT**」フィールドと同じ形式です。接尾語 **w** および **s** は、このフィールドでは無効です。

LETTER/S

このルールが有効な場合に使用される、タイムゾーンの省略形の 可変部 を指定します。このフィールドに **-** (ハイフン) が含まれると、可変部は **null** になります。**S** 文字は **EST** を表すために使用され、**D** 文字は **EDT** を表すために使用されます。

ゾーン行は次のような形式です。

```
Zone NAME GMTOFF RULES/SAVE FORMAT [UNTIL]
```

ゾーン行を構成するフィールドは下記のとおりです。

NAME

タイムゾーンの名前を指定します。これは、ゾーン用の時刻変換情報ファイルを作成するために使用される名前です。

GMTOFF

このゾーンで標準時間を得るために GMT へ追加する時間量を指定します。このフィールドは、ルール行の「AT」や「SAVE」フィールドと同じ形式です。GMT から時間を減算する必要がある場合は、フィールドの最初に負符号を付けてください。

RULES/SAVE

タイムゾーンに適用するルール名を指定するか、あるいは代わりに、ローカル標準時間に追加する時間量を指定します。このフィールドの値が - (ハイフン) である場合は、標準時間が常にタイムゾーンに適用されます。

FORMAT

このタイムゾーンで使用される、タイムゾーンの省略形を指定します。 %s 文字は、タイムゾーンの省略形の可変部の場所を示すために使用されます。

UNTIL

GMT オフセットまたはルールが地域によって変わる時刻を指定します。年、月、日、時刻で指定します。これが指定されると、指定された GMT オフセットおよびルールの変更から始まり、指定された時刻にいたるまでの、タイムゾーン情報が作成されます。

次の行は 継続行 である必要があります。継続行は、前の行で使用したファイルへ、前の行の **UNTIL** フィールドで指定した時刻から始まる情報を入れます。この行は、ゾーン行と同じ形式ですが、Zone という文字列と、名前が省略されているところが異なります。継続行には、ゾーン行と同様、 **UNTIL** フィールドを指定することができ、次の行もさらに継続であることを示します。

リンク行は次のような形式です。

Link LINK-FROM LINK-TO

LINK-FROM フィールドは、ゾーン行の **NAME** フィールドと同様に表示されます。**LINK-TO** フィールドは、そのゾーンの代替名として使用されます。

継続行以外は、行は、どの順に入力してもかまいません。

zic コマンドは、1901 年 12 月 14 日より前の日付を含むコンパイル入力を制限しています。これより前の日付は、32 ビット **time_t** データ・タイプでは表現できないからです。

フラグ

項目	説明
-d <i>Directory</i>	/usr/share/lib/zoneinfo/ 標準ディレクトリーの代わりに、 <i>Directory</i> ディレクトリーへ時刻変換情報ファイルを作成します。
-l <i>TimeZone</i>	<i>TimeZone</i> タイムゾーンをローカル時間として使用します。 zic コマンドは、ファイルが下記のようなリンクを含むかのように動作します。 Link <i>timezone localtime</i>
-L <i>Leapseconds</i>	leapseconds ファイルからうるう秒情報を読み取ります。このオプションが使用されていない場合、うるう秒情報は出力に表示されません。
-p <i>Posixrules</i>	POSIX 形式のタイムゾーン環境変数を扱う場合は、 <i>posixrules</i> 規則を使用します。 zic コマンドは、ファイルが次の例のようなリンクを含んでいるかのように動作します。 Link <i>timezone posixrules</i>
-v	データ・ファイル内に現れる年が、システム時刻の値 (0:00:00 AM GMT, January 1, 1970, to 3:14:07 AM GMT, January 19, 2038) で表される年の範囲外である場合のメッセージを指定します。
-y <i>YearIsType</i>	年のタイプを検査するために使用される場合、 /usr/sbin/yearistype コマンドではなく、指定した yearistype コマンドを使用します。

パラメーター

項目	説明
<i>FileName</i>	作成される時刻変換情報ファイルを指定する入力行を含むファイル。 <i>FileName</i> が - (ハイフン) である場合は、標準入力を読み取られます。

例

1. ルール行は次のような形式です。

```
Rule USA 1970 max - Sep Sun<=14 3:00 0 S
```

2. ゾーン行は次のような形式です。

```
Zone Turkey 3:00 Turkey EET%s
```

3. リンク行は次のような形式です。

```
Link MET CET
```

4. 入力タイムゾーン情報を含むファイル **timezone.infile** をコンパイルし、バイナリーを標準タイムゾーン・ディレクトリー **/usr/share/lib/zoneinfo/** 内へ入れるには、下記のように入力します。

```
zic timezone.infile
```

5. 入力タイムゾーン情報を含むファイル **timezone.infile** をコンパイルし、バイナリーを **-d** オプションで指定したディレクトリーへ入れるには、下記のように入力します。

```
zic -d tzdir timezone.infile
```

6. 年の範囲が間違っている場合に、タイムゾーンの入力ファイルをコンパイル中に警告をレポートするには、下記のように入力します。

```
zic -v timezone.infile
```

7. 年のタイプを検査するために、**-y** フラグで指定した **yearistype** ファイルを使用して、入力タイムゾーン情報を含むファイル **timezone.infile** をコンパイルするには、下記のように入力します。

```
zic -y year timezone.infile
```

終了状況

0 コマンドは正常に完了しました。

>0 エラーが発生しました。

ファイル

項目	説明
/usr/sbin/yearistype	年の型を検査するために使用される yearistype コマンドが入っています。
/usr/sbin/zic	SystemV zic コマンドが入っています。
/usr/share/lib/zoneinfo	zic コマンドで作成されたファイルに使用する標準ディレクトリー。

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。これらの名称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・オファリング」）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・オファ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・オファリング」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・オファリング」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・オファリング」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・オファリング」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。

Windows は、Microsoft Corporation の米国およびその他の国における商標です。

UNIX は、The Open Group の米国およびその他の国における登録商標です。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アカウントティング・システム

フォーマットの問題の訂正 140

イベント応答リソース・マネージャー (ERRM)

コマンド

wallevent 86

スクリプト

wallevent 86

イメージ

ダンプ

xwd コマンドを使用する 269

取り出し

xwud コマンドを使用する 270

表示

xwud コマンドを使用する 270

[カ行]

解析プログラム

yacc コマンドによる作成 273

仮想キーの割り当て

構成

xmbind コマンドを使用する 200

仮想フレーム・バッファ (VFB)

X サーバーの始動 143

仮想メモリー

統計情報の報告 50

キーの割り当て

仮想を構成する

xmbind コマンドを使用する 200

キーワード

SCCS

値の置き換え 9

繰り返される応答

生成 275

ゲーム

三目並べ 141

権限情報

編集および表示

xauth コマンドを使用 162

構成

仮想キーの割り当て

xmbind コマンドを使用する 200

コマンド

イベント応答リソース・マネージャー (ERRM)

wallevent 86

vacation 1

vpdadd 64

vsdchgserver 69

vsdelnode 70

vsdnode 73

vsdsklst 75

w 83

wall 85

wallevent 86

who 102

wlmassign 111

wlmcntrl 115

wlmstat 118

wparprnterr 134

X 143

xargs 158

xdm 168

xmbind 200

xmodmap 206

xntpd 216

xterm 243

ypcat 277

ypinit 279

yppasswd 281

yppoll 284

yppush 285

ypset 287

ypwhich 290

ypxfr 291

[サ行]

シェル

C 内の実行可能ファイルの位置決め 100

シェル・スクリプト

繰り返される応答

生成 275

時刻

現在の時刻を表示する

xclock コマンドを使用する 165

スクリーン・カラー特性データ

照会

xcmsdb コマンドを使用する 167

ルート・ウィンドウから照会する

xcmsdb コマンドを使用する 167

ルート・ウィンドウから除去する

xcmsdb コマンドを使用する 167

スクリーン・ロック
制御
 xss コマンドを使用する 240
スクリプト
 イベント応答リソース・マネージャー (ERRM)
 wallevent 86
 wallevent 86
セキュリティ
 ワークステーション画面のロック
 xss コマンドを使用する 240

[タ行]

ダンプ・ファイル
 プリンター出力用のフォーマット
 xpr コマンドを使用する 224
端末エミュレーター
 X Window System 用に提供する
 xterm コマンドを使用する 243
通信チャンネル
 セキュアでのメールの受信
 xget コマンドを使用する 187
 セキュアでメールを送信
 xsend コマンドを使用する 234
デーモン
 ypbind 276
 yppasswdd 282
 ypserv 286
 ypupdated 289

[ハ行]

パス名
 実行可能ファイル
 検索 100
ファイル
 圧縮 296
 実行可能
 位置決め 100
 セクションの位置探索 98
 展開 295
 SCCS
 識別情報の表示 92
 妥当性検査 3
フォント
 X Window のディスプレイ・サーバーへの提供 184
フォント・サーバー
 追加 155
フォント・ディレクトリー
 追加 156
フォント・パス・エレメント
 除去 157
プログラム
 モニター
 watch コマンドの使用 88

プログラム (続き)
 Imakefile からの Makefile の作成 201
プログラム・リスト
 フォーマット
 vgrind コマンドを使用する 12
プロセス中断
 実行の中断 84
別名
 位置決め 100
ヘルプ
 コマンドの機能の説明 93
ホスト・マシン
 アクセスの制御
 xhost コマンドを使用する 189
ボリューム・グループ
 活動開始
 varyonvg コマンドを使用する 6
 活動停止
 varyoffvg コマンドを使用する 4

[マ行]

メール
 セキュア通信チャンネルでの受信
 xget コマンドを使用する 187
 セキュア通信チャンネルでの送信
 xsend コマンドを使用する 234
メッセージ
 他のユーザーへの送信
 write コマンドを使用する 134
 の後処理のためのプロンプト
 whatnow コマンドを使用する 94
 の受信側のアドレスの検査
 whom コマンドを使用する 109
 の受信側のアドレスのリスト
 whom コマンドを使用する 109
 リモート・システムからの受信
 writesrv コマンドを使用する 139
 リモート・システムからの送信
 writesrv コマンドを使用する 139
メモリー管理
 仮想メモリー統計情報の報告 50
文字列、共用
 xstr コマンドを使用する 241

[ヤ行]

ユーザー
 ログインしているユーザーを識別する
 who コマンドを使用する 102
 ログイン名の表示 105
ユーザー名ディレクトリー
 ニックネームの検索
 whois コマンドを使用する 107

ユーザー名ディレクトリー (続き)
ID の検索
 whois コマンドを使用する 107

[ラ行]

リアルタイム・メッセージ
 他のユーザーへの送信
 write コマンドを使用する 134
リソース・データベース
 X サーバー用のユーティリティーの実行
 xrdp コマンドを使用する 231
ログイン、ユーザー
 識別する
 who コマンドを使用する 102
ロックする、X ディスプレイを、パスワードが入力されるまで
 xlock コマンドを使用する 197

[ワ行]

ワークステーション画面
 ロック
 xss コマンドを使用する 240
割り当て
 仮想キーを構成する
 xmbind コマンドを使用する 200

A

ARPANET
 whois コマンドのユーザー 107
ATE プログラム
 xmodem コマンド 203
 xmodem によるファイルの転送 203

E

ERRM コマンド
 elogevent 86
ERRM スクリプト
 wallevent 86

M

MH コマンド
 で使用するビジュアル・インターフェースの起動
 vmh コマンドを使用する 34

N

network information service 276
Network Time Protocol コマンド
 xntpd 213

NIS コマンド
 ypcat 277
 ypinit 279
 ypmatch 280
 yppasswd 281
 yppoll 284
 yppush 285
 ypset 287
 ypwhich 290
 ypxfr 291
NIS デモン
 ypbind 276
 yppasswdd 282
 ypserv 286
 ypupdated 289

S

SCCS
 キーワード
 値の置き換え 9
 ファイル
 識別情報の表示 92
 妥当性検査 3
SCCS コマンド
 val 3
 vc 9
 what 92

V

vacation コマンド 1
varyoffvg コマンド 4
varyonvg コマンド 6
vgrind コマンド 12
vi エディター
 始動 33
vi コマンド 14
view コマンド 33
vmh コマンド 34
vmo コマンド 35
vpdadd コマンド 64
vsdata1st コマンド 67
vsdchgserver コマンド 69
vsdelnode コマンド 70
vsdnode コマンド 73
vsdsklst コマンド 75
vsdvgtg コマンド 79

W

w コマンド 83
wall コマンド 85
wallevent コマンド 86
wallevent スクリプト 86

watch コマンド 88
wc コマンド 90
whatis コマンド 93
whatnow コマンド 94
which_fileset コマンド 100
who コマンド 102
whodo コマンド 105
whois コマンド
 の説明 107
 の例 108
 ARPANET
 での使用 107
wlmassign コマンド 111
wlmcheck コマンド 113
wlmcntrl コマンド 115
wlmstat コマンド 118
wol コマンド 123
wparerr コマンド 125
wparexec
 作成する、アプリケーション・ワークロード・パーティショ
 ン または仕様ファイル 126
wparprnterr コマンド 134
wpar_reg_script 133
write コマンド 134
writesrv コマンド 139
wtmpfix コマンド 140

X

x コマンド 143
X サーバー
 始動 143
X ディスプレイ
 上に troff ファイルを表示
 xpreview コマンドを使用する 226
 パスワードが入力されるまでロック
 xlock コマンドを使用する 197
X レンダー拡張機能
 X サーバーの始動 143
xargs コマンド 158
xauth コマンド 162
xclock コマンド 165
 デフォルトを設定する 166
xcmsdb コマンド 167
xdm コマンド 168
xfs コマンド 184
xget コマンド 187
xhost コマンド 189
xinit コマンド 190
xlock コマンド 197
xlsfonts コマンド 199
xmbind コマンド 200
xmodem コマンド 203
xmodem プロトコル 203
xmodmap コマンド 206
xmpeek コマンド 208

xmscheck コマンド 210
xmtopas コマンド 211
xmwlm コマンド 202
xntpd デーモン 213
xntpd コマンド 216
xpr コマンド 224
xpreview コマンド 226
xprofiler 229
xrdb コマンド 231
xsend コマンド 234
xsetroot コマンド 239
xss コマンド 240
xstr コマンド 241
xterm コマンド 243
 端末エミュレーションを提供する
 X Window システム用 243
xwd コマンド 269
xwud コマンド 270
X-Window
 環境オプションの設定 235
 ルート・ウィンドウ・パラメーターの設定 239
x_add_fs_fpe コマンド 155
x_add_nfs_fpe コマンド 156
x_rm_fpe コマンド 157

Y

yacc コマンド 273
ypbind デーモン 276
ypcat コマンド 277
ypinit コマンド 279
ypmatch コマンド 280
yppasswd コマンド 281
yppasswdd デーモン 282
yppoll コマンド 284
yppush コマンド 285
ypserv デーモン 286
ypset コマンド 287
ypupdated デーモン 289
ypwhich コマンド 290
ypxfr コマンド 291

Z

zcat コマンド 295
zdump コマンド 296
zic コマンド 298



Printed in Japan