



# Windows コマンドラインの 使い方

## この節の内容

|                          |      |
|--------------------------|------|
| • コマンドプロンプトの概要           | P.30 |
| コマンドプロンプトとは              | P.30 |
| 起動と終了                    | P.31 |
| カレントディレクトリ               | P.31 |
| Path (コマンドサーチパス)         | P.33 |
| コマンドラインの編集とコマンド履歴 (ヒストリ) | P.34 |
| ファイル名の補完                 | P.34 |
| パイプとリダイレクト               | P.35 |
| コラム: コマンドプロンプト画面の編集      | P.36 |
| • Windows のコマンド          | P.37 |
| 基本コマンド                   | P.37 |
| コマンドの調べ方                 | P.38 |
| Windows のヘルプ             | P.39 |
| • doskey の活用             | P.41 |
| doskey の起動と終了            | P.41 |
| マクロの登録                   | P.41 |
| マクロファイル                  | P.43 |
| doskey の自動実行             | P.43 |
| コラム: マクロとバッチファイル         | P.45 |



## コマンドプロンプトの概要

Windows では、ほとんどの作業を GUI ベースで行えるようになっていますが、コマンドプロンプトも併用することで、使い勝手を向上させることができます。

Linux との混在環境を考える場合、コマンドプロンプトの活用は大変重要です。

### コマンドプロンプトとは

コマンドプロンプトは、コマンドを入力するためのウィンドウです。

#### コマンドプロンプト

dir コマンド (ファイル一覧を表示するコマンド) を実行

コマンドやファイル名の大小文字は区別されない

dir コマンドの  
実行結果が  
表示される

notepad (メモ帳)  
を起動

write (ワードパッド)  
を起動

```

C:\WINDOWS\System32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\nisi>dir
ドライブ C のボリューム ラベルがありません。
ボリューム シリアル番号は 80D1-DF58 です

C:\Documents and Settings\nisi のディレクトリ

2002/12/09  21:25    <DIR>          .
2002/12/09  21:25    <DIR>          ..
2002/12/09  17:27    <DIR>          Favorites
2002/12/09  17:55    <DIR>          My Documents
2002/08/14  23:45    <DIR>          スタート メニュー
2002/12/14  22:12    <DIR>          デスクトップ
                0 個のファイル                0 バイト
                6 個のディレクトリ  2,476,134,400 バイトの空き領域

C:\Documents and Settings\nisi>notepad
C:\Documents and Settings\nisi>write
C:\Documents and Settings\nisi>

```

“notepad” や “write” などのプログラム名は、スタートメニューで確認できます。スタートメニューでコマンドラインから起動したいプログラムを右クリックし、「プロパティ (P)」を選択して「ショートカット」ページを開きます。「リンク先 (T)」に書かれているのが実行プログラム名です。

ユーザーの入力を受け取り、OS (カーネル) に渡す役割を受け持つプログラムをシェル (shell) といいます。Windows XP や 2000 では CMD.EXE が、Windows 95/98/Me では command.com が標準のシェルとして使用されます。

▶ bash や tcsh を使用する : 本文 P.341

プロンプト (prompt) とは、ユーザーに入力や操作を促すメッセージのことで、Windows では一般に “>” 記号が使用されます。また、プロンプトの脇 (入力位置) には、カーソル (ラテン語で「走者」の意) が表示されます。

**Tips** プロンプトは、環境変数 prompt で変更可能です。デフォルトでは「\$P\$G」が指定されています。\$P はカレントディレクトリ (P.31) を、\$G は“>”(greater) 記号を意味しています (「help prompt」参照)

注意

ここでは、Windows XP (Home Edition) のコマンドプロンプトを中心に解説しています。デフォルトの設定内容や、収録されているコマンドは Windows の種類やバージョン、インストール時の指定によって異なることがあります。

## 起動と終了

コマンドプロンプトは、スタートメニューの「プログラム (P)」「アクセサリ」「コマンドプロンプト」で起動します。また、「ファイル名を指定して実行」で「cmd」と入力しても起動できます。

「exit」で終了します。

**Tips** Windows 95/98/Me では command と指定します。command (command.com) は 16bit ベースのシェルです。command は Windows 2000/XP でも使用できますが、cmd (CMD.EXE) のほうが高機能です。

## カレントディレクトリ

現在作業しているディレクトリのことを、カレントディレクトリ (current directory) といいます。カレントディレクトリにあるファイルは、ディレクトリ名を指定せずに使用できます。

なお、Windows のヘルプなどでは、カレントディレクトリが“現在のディレクトリ”と表現されています。ここでは、Linux などでも広く使用される用語として「カレントディレクトリ」を使用しています。

### デフォルトのカレントディレクトリ

Windows XP の場合、コマンドプロンプトを起動したときのカレントディレクトリは“C:\Documents and Settings\ユーザー名”となっています。別のディレクトリにしたい場合は、コマンドプロンプトのプロパティで変更します。

▶ 設定例：次ページ

**Tips** 起動時のカレントディレクトリは、環境変数 HOMEDRIVE (インストールドライブ。通常は C:) と環境変数 HOMEPATH (Documents and Settings\ユーザー名) の組み合わせで設定されています。

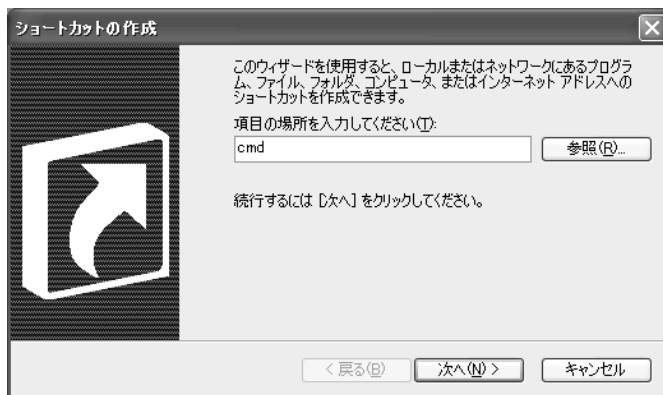
スタートメニューの  
「コマンドプロンプ  
ト」で右クリック  
「プロパティ (P) 」

作業フォルダを変更する



よく使うディレクトリを設定したショートカットを作っておいてもよいでしょう。  
デスクトップで右クリックし、「新規作成 (W) 」 「ショートカット (S) 」を選択しま  
す。ショートカットの作成ダイアログが開くので、“ 項目の場所 ”(ここでは実行コマン  
ド名のこと)としてcmdを指定します。

ショートカットの  
作成



作成したショートカットのプロパティで、作業フォルダを変更します。

## Path(コマンドサーチパス)

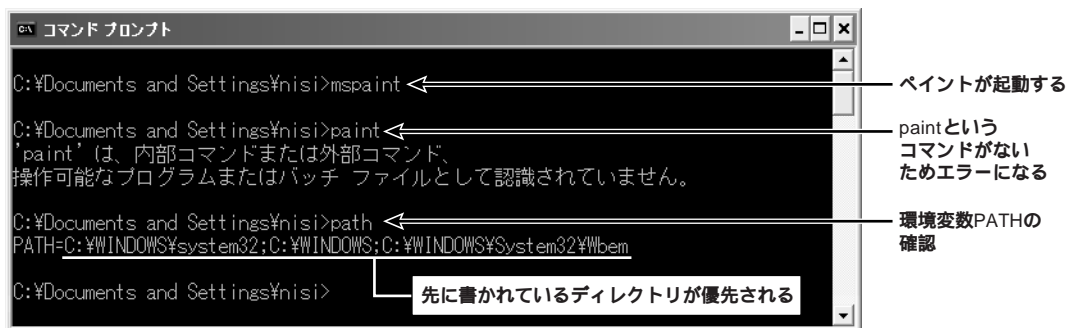
カレントディレクトリおよび環境変数Pathに設定されているディレクトリにある実行ファイルは、ファイル名だけで実行できます。ディレクトリが環境変数Pathに指定されていることを、“Pathが通っている”と表現することがあります。

Pathの値は、pathコマンドで確認できます。デフォルトでは、C:¥WINDOWS¥system32 及びC:¥WINDOWSが指定されています。

### Tips

環境によっては、すべて大文字で“PATH”と定義されていることがあります。Windows では環境変数の大小文字が区別されません。

#### 環境変数PATHの確認



優先順位：(1) cmd.exeの内部コマンド (dirやcopyなど、CMD.EXEが内蔵しているコマンド)  
(2) カレントディレクトリ  
(3) PATHに指定されたディレクトリ

### Tips

Windows の場合、実行ファイル名の拡張子を省略できます。同じディレクトリに同名のコマンドがある場合、優先順位は.com、.exe、.bat、.cmdの順になります。

#### Pathの変更

環境変数Pathは、システムのプロパティで設定します。

▶ 実行画面：次ページ

コントロールパネルの「システム」(カテゴリ：パフォーマンスとメンテナンス)で「システムのプロパティ」を開きます。「詳細設定」ページの[環境変数(N)]ボタンをクリックすると「環境変数」ダイアログが開くので、ここで設定します。

システムのプロパティ 詳細設定  
環境変数



Pathを選択して[編集 (I)]をクリック

**Tips** システムのプロパティは、マイ コンピュータで右クリック 「プロパティ (R)」でも開くことができます。

## コマンドラインの編集とコマンド履歴(ヒストリ)

コマンドラインでは、左右の矢印キーでカーソルを動かして入力内容を修正できます。

また、コマンドプロンプトの画面を開いてから入力したコマンドは、上下矢印キーで呼び出すことができます。また、[F7]キーで一覧表示して選択することも可能です。ただし、コマンドを一度も入力していない場合はコマンド履歴ウィンドウは開きません。

**Tips** command.com の場合は doskey ( P.37 ) を併用することでコマンドラインの編集と履歴の使用が可能になります。

## ファイル名の補完

CMD.EXE は、[Tab]キーによるファイル名の補完機能をサポートしています。補完機能とは、ファイル名の入力を補う機能で、たとえば「a」と入力して[Tab]キーを押すと、aから始まるファイルが順次表示されます。

補完機能で使用するキーは、レジストリで変更できます。レジストリは、regedit コマンド ( regedit.exe ) または reg コマンド ( reg.exe ) で編集できます。危険が伴うので、Windows に不慣れな場合には変更はお勧めしません。

▶ reg コマンド使用  
例 ( 画面 ) : 本文  
P.5

## レジストリの修正箇所

[HKEY\_LOCAL\_MACHINE¥Software¥Microsoft¥Command Processor]  
(システム全体用)

または

[HKEY\_CURRENT\_USER¥Software¥Microsoft¥Command Processor]  
(ユーザー用：システム全体用の設定より優先される)

## レジストリのキーと設定値

"CompletionChar"=dword:00000009  
キー 設定値

- ・設定値：0 で無効、9 で[Tab]([Ctrl]+[I])、4 で[Ctrl]+[D]など

初期値では、HKEY\_LOCAL\_MACHINEの設定が64(スペースキー：無効)、HKEY\_CURRENT\_USERの設定が9([Tab],[Ctrl]+[I])となっています。また、Windows 2000では、デフォルトの設定値が0(無効)となっています。

**Tips**

「cmd /F:ON」で、[Ctrl]+[F]でファイルとディレクトリ、[Ctrl]+[D]でディレクトリのための補完という使い分けが可能になります。

**パイプとリダイレクト**

コマンドが出力するメッセージは、「> ファイル名」で保存したり、「| コマンド名」で別のコマンドに渡したりできます。>をリダイレクト、|をパイプといいます。

なお、「> ファイル名」で既存のファイルを指定すると、元の内容が破棄されて新しいファイルが作成されます。既存の内容に追加したい場合は「>> ファイル名」のように指定します。

## リダイレクトとパイプの例

## Windows

```
>dir > dirlist.txt      dir コマンドの実行結果を dirlist.txt に保存
>dir >> dirlist.txt     dir コマンドの実行結果を dirlist.txt に追加
>dir | more             dir コマンドの実行結果を more コマンドに渡す
dir : ファイルの一覧を表示するコマンド
more : テキストデータを1画面ずつ表示するコマンド
```

**Tips**

CMD.EXE は、bash(Linux で使用されるシェル)のように、「1>」で標準出力、「2>」で標準エラー出力をそれぞれリダイレクトできます。

## Column

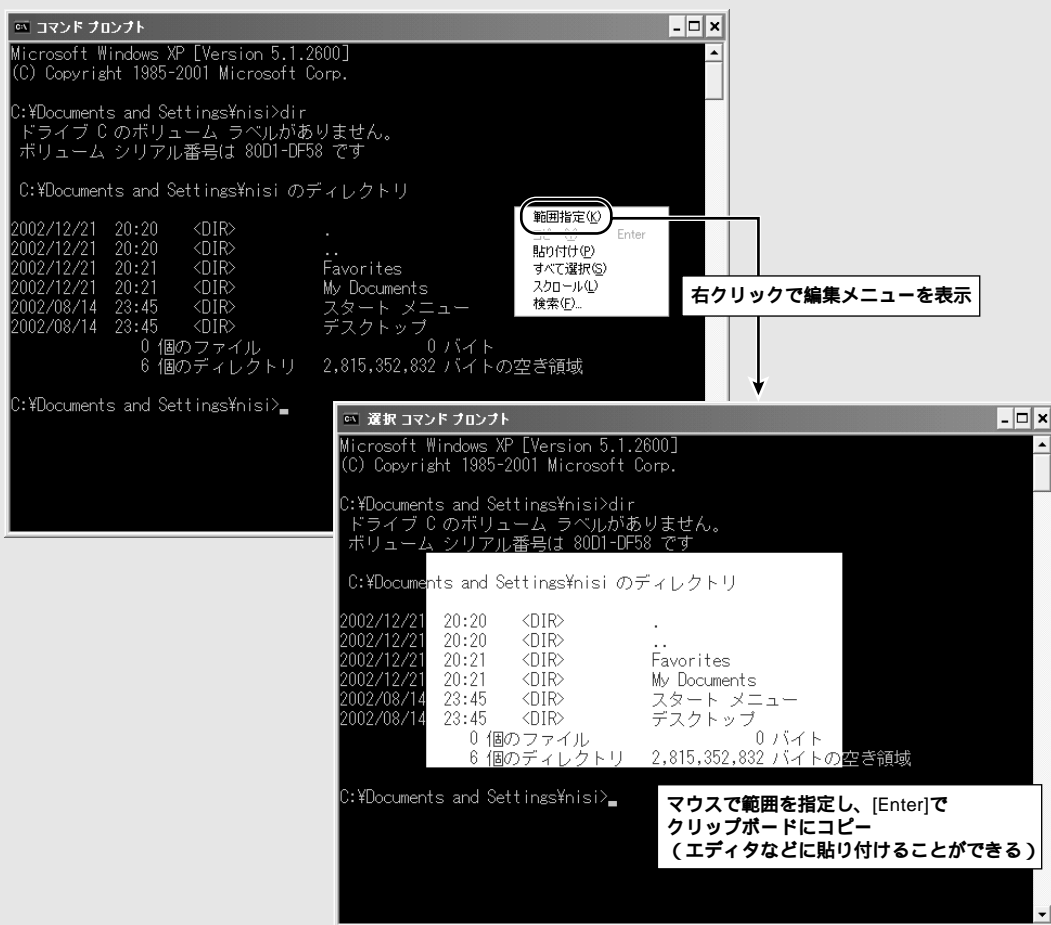
### コマンドプロンプト画面の編集

コマンドプロンプト画面に表示されている文字列をエディタなどにコピーしたい場合は、コマンドプロンプト画面で右クリックして編集メニューを表示します。

「範囲指定 (K)」で範囲指定モードになるので、マウスで範囲を選択します。範囲を指定して [Enter] キーを押すと選択範囲の内容がクリップボードにコピーされます。

また、編集メニューの「貼り付け (P)」で、クリップボードの内容をコマンドラインにコピーすることができます。

#### 編集メニューを表示



**Tips** コマンドプロンプトのプロパティの「オプション」ページで、「簡易編集モード (Q)」を有効にすると、編集メニューを使わずに範囲指定と貼り付け (右クリック) ができるようになります。





## Windows のコマンド

Windows 環境には、DOS 時代からのものも含んだ多くのコマンドが用意されています。

**Tips** ほとんどのコマンドは C:\WINDOWS\system32 に収録されています。

▶ ネットワーク用の  
コマンド：  
netbase.pdf P.20、  
本文 P.61

### 基本コマンド

Linux でよく使われるコマンドとの対応は次の通りです。

Linux コマンドと Windows コマンドの対応

| Linux | Windows       | 内容  |
|-------|---------------|---|
| ls    | dir           | ファイルの一覧を表示。「dir ファイル名」または「dir ディレクトリ名」で実行。ファイルやディレクトリはそれぞれ複数指定可能。ワイルドカードとして * と ? が使用可能。ls のようにファイル名のみの表示としたい場合は /B オプションを付ける |
| cd    | cd, chdir     | カレントディレクトリを移る。「cd ディレクトリ名」で指定。ディレクトリを指定しない場合、現在のカレントディレクトリを表示   |
| cp    | copy          | ファイルのコピー。「copy ファイル名 コピー先」で指定。コピー先にはファイル名またはディレクトリを指定。cp コマンドと違い、コピー先を省略するとカレントディレクトリにコピーされる                                  |
| mv    | move          | ファイルの移動。「move ファイル名 移動先」で指定。ディレクトリに対しても使用可能   |
| mv    | ren, rename   | ファイル名変更。「ren ファイル名 新しいファイル名」で指定。ディレクトリに対しても使用可能   |
| rm    | del           | ファイルを削除。「del ファイル名」で指定。ディレクトリを指定した場合はディレクトリ内のファイルを削除  |
| rmdir | rd, rmdir     | 空のディレクトリを削除。「rd ディレクトリ名」で指定   |
| mkdir | md, mkdir     | ディレクトリを作成。「md ディレクトリ名」で指定   |
| chmod | attrib        | ファイルの属性を変更。「attrib +R ファイル名」のように、「+ 属性」で属性を追加、「- 属性」で属性を削除する  |
| echo  | echo          | メッセージを表示。環境変数を参照する場合は「echo %PATH%」のように、% で囲む  |
| cat   | type          | テキストファイルの内容を表示。「type ファイル名」で指定。ファイルは複数指定可能  |
| more  | more          | テキストデータを 1 画面ずつ表示。「more ファイル名」または「type ファイル名 ! more」のように指定(使用例: P.38)   |
| grep  | find, findstr | テキストファイル内の文字列を検索。「find "検索文字列" ファイル名」で、複数のファイルを指定可能。行番号や正規表現が必要な場合は findstr を使用   |
| clear | cls           | 画面の消去   |
| -     | start         | 新しいコマンドプロンプトウィンドウを開く。なお、「start ディレクトリ名」でフォルダウィンドウが、「start ファイル名」で、ファイルに応じたプログラムが起動する。なお関連づけは assoc または ftype コマンドで確認・変更可能     |

## コマンドの調べ方

各コマンドの使い方は、/?オプションで確認できます。

/?オプションの表示例

```

C:\Documents and Settings\nisi>tree /?
ドライブやパスのフォルダ構造を図式表示します。

TREE [ドライブ:] [パス] [/F] [/A]

    /F  各フォルダのファイル名を表示します。
    /A  拡張文字ではなく、ASCII 文字で表示します。

C:\Documents and Settings\nisi>

```

**Tips** ヘルプメッセージが長い場合は、「コマンド名 /? | more」のように、more コマンドを使って表示するとよいでしょう。

### どのようなコマンドがあるのか

help コマンドで、主要コマンドの一覧を表示できます。1画面では表示しきれません。「help | more」のように、more コマンドを使うと1画面ずつ表示できます。

help コマンド

```

C:\Documents and Settings\nisi>help | more
特定のコマンドの詳細情報は、「HELP コマンド名」を入力してください
ASSOC  ファイル拡張子の関連付けを表示または変更します。
AT      コマンドやプログラムを指定した日時に実行します。
ATTRIB  ファイルの属性を表示または変更します。
BREAK   拡張 CTRL+C チェックを設定または解除します。
CAcls   ファイルのアクセス制御リスト (ACL) を表示または変更します。
CALL    バッチ ファイル中から、別のバッチ ファイルを呼び出します。
CD      現在のディレクトリを表示または変更します。
CHCP    有効なコード ページ番号を表示または設定します。
CHDIR   現在のディレクトリを表示または変更します。
-- More

```

スペースで次の1画面分を表示。  
[Enter]で次の1行を表示

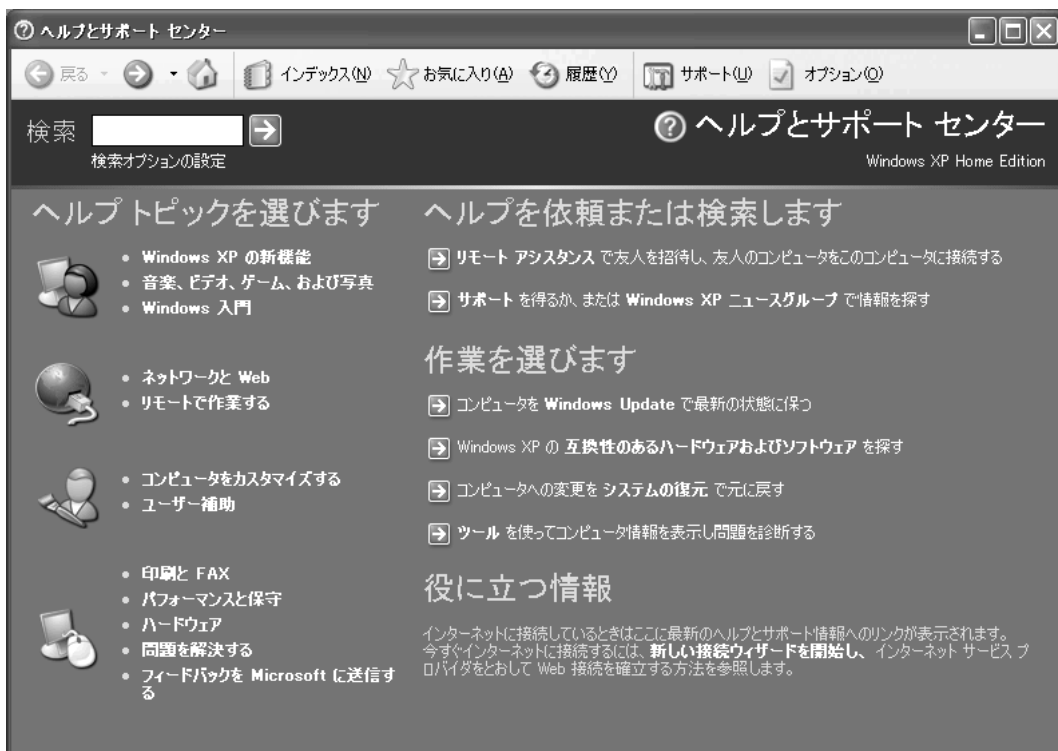
## Windows のヘルプ

コマンドの使い方は、Windows XP の「ヘルプとサポートセンター」でも調べることができます。

「ヘルプとサポートセンター」は[スタート] 「ヘルプとサポート(H)」で起動します。

**Tips** ヘルプとサポートセンターの実行プログラムは C:\WINDOWS\PCHEALTH\HELPCTR\Binaries\helpctr.exe です。

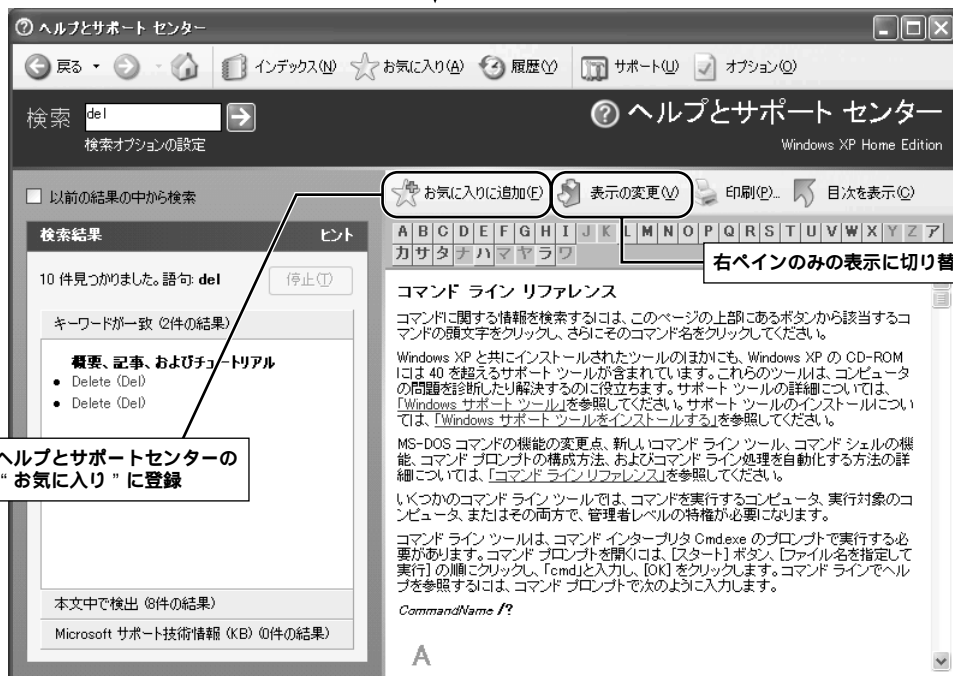
### ヘルプとサポートセンター



検索ボックスにコマンド名を入力して[Enter]



検索結果に表示されたタイトルをクリックすると、右ペインにヘルプが表示される



ヘルプとサポートセンターの  
“お気に入りに”に登録

右ペインのみの表示に切り替える



## doskey の活用

doskey は、コマンドラインを拡張するツールです。コマンドラインの編集や再呼び出し(履歴) マクロ(エイリアス)の使用などが可能になります。

なお、Windows XP で使用される CMD.EXE は、標準でコマンドラインの編集や再呼び出しをサポートしています。したがって、doskey は主にマクロ機能のために使用されます。

### doskey の起動と終了

▶ doskey の自動起動 : P.43

「doskey」で起動します。メッセージなどは表示されませんが、一度起動すると、コマンドプロンプトで doskey の機能が使用できるようになります。

```
>doskey
```

「exit」で cmd とともに終了します。

### マクロの登録

▶ マクロの保存 : P.43

「doskey マクロ名=コマンド」で登録します。たとえば、clear という名前で cls (画面をクリアするコマンド。ClearScreen) を実行したい場合は次のようにします。

マクロの登録例

Windows

```
>doskey clear=cls
```

### 引数を使う

引数は \$1、\$2... で表します。\$1 が 1 つ目の引数、\$2 が 2 つ目の引数にそれぞれ対応し、\$9 まで使用できます。「すべての引数」は \$\* で表します。

### Tips

\$1、\$2... は、BAT ファイルの %1、%2 に相当します。

たとえば、cd コマンドの代わりに pushd コマンドを使う場合は次のようにします。pushd は、現在のディレクトリ名を記憶した上でディレクトリを移動するコマンドで、popd コマンドで元のディレクトリに戻ることができます。

```
>doskey cd=pushd $1
```

**Tips** pushd・popd コマンドは、bash や tcsh でも使用できます。なお、Linux で使用される cd コマンドは「cd -」で直前のディレクトリに戻ることができます。

なお、コマンド名と同じ名前のマクロを登録した場合、コマンドラインで「cd」のように先頭に空白を入れることで、元のコマンドを実行できます。

cd と cd- で pushd/popd を実行

マクロ名「cd」で  
「pushd 引数」を  
実行するように設定

マクロ名「cd-」で  
「popd」を実行する  
ように設定

「pushd %windows」  
が実行される

「popd」が実行される  
(直前のディレクトリ  
に戻る)

先頭に空白を入れる  
ことで、元のcdコマ  
ンドが実行できる

```

C:\> doskey cd=pushd %1
C:\> doskey cd-=popd
C:\> cd %windows
C:\WINDOWS> cd-
C:\Documents and Settings\%nisi> cd %windows
C:\WINDOWS>
  
```

### 複数のコマンドを実行する

1つのマクロで複数のコマンドを実行したい場合は、\$Tで区切ります。

たとえば、filefindという名前で、“指定ディレクトリに移動して、「dir /s /b ファイル名」を実行し、元のディレクトリに戻る”という作業をするには、次のように設定します。

複数のコマンドを登録する例

Windows

```
>doskey filefind=pushd %2 $t dir /s /b %1 $t popd
```

「filefind ファイル名 ディレクトリ名」で、指定ディレクトリ下のファイルを探すことができる  
(実行例：P.45)

dir /s /b : /sはサブディレクトリ下も検索するオプション、/bはファイル名のみを表示するオプション。なお、/pオプションを併用すると、ファイルのリストを1画面ずつ表示できます。

**Tips** コマンドラインでは、「pushd c:\windows && dir /s /b net.exe && popd」のように、&&で区切ります。

このほか、以下の特殊文字が用意されています。

doskey で使用できる特殊文字

|                 |                      |
|-----------------|----------------------|
| \$G             | > (出力のリダイレクト)        |
| \$G\$G          | >> (出力の追加リダイレクト)     |
| \$L             | < (入力のリダイレクト)        |
| \$B             | ! (パイプ) 使用例: 本文P.339 |
| \$T             | コマンドの区切り (&&)        |
| \$ \$           | \$記号自身               |
| \$1 ~ \$9, \$ * | コマンドライン引数            |

## マクロファイル

「doskey /macrofile=ファイル名」で、マクロを書いたファイルを読み込むことができます。

```
>doskey /macrofile=macros.txt
```

### 定義済みマクロの表示と保存

▶ リダイレクト:  
P.35

「doskey /macros」で、現在定義されているマクロが表示できます。保存しておきたい場合は、リダイレクトでファイルに出力します。

```
>doskey /macros          定義済みマクロを表示
>doskey /macros > macros.txt  定義済みマクロを macros.txt に保存(>の代わりに
                                >>で指定すると、ファイルに追加される)
```

## doskey の自動実行

▶ 設定画面: 次ページ

doskey を常に使用したい場合は、コマンドプロンプトのプロパティで、cmd.exe の引数に doskey を追加します。

```
%SystemRoot%\system32\cmd.exe /k "doskey /macrofile=c:\macros.txt"
```

コマンドプロンプトでdoskeyを常に起動する



cmd.exeの後に、「/k doskey」を追加する。  
マクロファイルを読み込ませたい場合は、/macrofile=オプションを指定

このほか、レジストリのAutoRunに指定して、システム起動時のdoskeyを起動しておく方法もあります。Windows 95/98/Meの場合はC:\autoexec.batに追加しておいてもよいでしょう。

レジストリは、regeditコマンド ( regedit.exe ) または reg コマンド ( reg.exe ) で編集できます。危険が伴うので、Windows に不慣れな場合には変更はお勧めしません。

レジストリの修正箇所

[HKEY\_LOCAL\_MACHINE\Software\Microsoft\Command Processor]

または

[HKEY\_CURRENT\_USER\Software\Microsoft\Command Processor]

レジストリのキーと設定値

"AutoRun" = "doskey /macrofile=c:\%macros.txt"

キー

設定値



## Column

## マクロとバッチファイル

Windows 環境( DOS 環境 )では、doskey によるマクロのほかにバッチファイルも使用できます。バッチファイルは、コマンドラインで実行する手順を書いたテキストファイルのことで、拡張子に“.BAT ”または“.CMD ”を指定します。複数のコマンドを実行したい場合は、バッチファイルが便利です。

**Tips**

複雑な制御を行いたい場合は WSH( Windows Scripting Host )を使用します。WSH は JavaScript や VBScript でスクリプトを書くことができ、ウィンドウの制御も可能です。

たとえば P.42 の filefind マクロですが、単に pushd と dir コマンドを実行するだけでは、移動先のディレクトリ名が表示されるため実行結果が見づらくなります。そこで、echo off を実行してから pushd,dir,popd を実行し、echo on を実行するように改善してみます。

## 改善した filefind マクロ

Windows

```
>doskey filefind=echo off $t pushd $2 $t dir /s /b $1 $t popd $t echo on
```

filefind マクロに  
echo off を追加

pushdコマンドで移動した際にプロンプトが表示されている

```

C:\Documents and Settings\nisi>doskey filefind=pushd $2 $t dir /s /b $1 $t popd $t echo on
C:\Documents and Settings\nisi>filefind *hosts* c:\windows
C:\WINDOWS>C:\WINDOWS\system32\drivers\etc\hosts
C:\WINDOWS\system32\drivers\etc\hosts.sam
C:\WINDOWS>
C:\Documents and Settings\nisi>doskey filefind=echo off $t pushd $2 $t dir /s /b $1 $t popd $t echo on
C:\Documents and Settings\nisi>filefind *hosts* c:\windows
C:\WINDOWS\system32\drivers\etc\hosts
C:\WINDOWS\system32\drivers\etc\hosts.sam
C:\Documents and Settings\nisi>

```

dirコマンドの実行結果だけが表示されている

しかし、マクロがあまり長いと、処理の内容がわかりにくく、後で修正するのが難しくなります。このような場合は、バッチファイルのほうが見やすく、メンテナンスも容易です。

さきほどの filefind マクロをバッチファイルで書き直すと次のようになります。なお、echo off をバッチファイル内で実行した場合は、バッチファイル実行中のみ有効となるため echo on は不要です。

#### filefind.bat の例

Windows

```
@echo off
pushd %1
dir /s /b /p %1
popd
```

また、バッチファイルでは、条件によって処理を中断したり、ラベルを使用したりすることが可能です。

filefind.bat を加工し、「引数が 1 つも指定されていなかったら使用方法を表示して終了」「2 つ目の引数(ディレクトリ名)が指定されていなかったら“C:¥”が指定されたものとする」という機能を追加してみましょう。次のようになります。

#### 改良した filefind.bat の例

Windows

```
@echo off
if "%1" == "" (
    echo %0 ファイル名 [ディレクトリ名]
    goto EXT
)
set destdir=%2
if "%destdir%" == "" set destdir=c:¥
pushd %destdir%
dir /s /b /p %1
popd
:EXT
```

ラベル(EXT)を使わずにバッチファイルを終了する場合は、goto EXT の代わりに“exit /b”とします。この場合は最終行のラベル(:EXT)は不要です。

「2 つ目の引数(ディレクトリ名)が指定されていなかったらカレントディレクトリを対象とする」という場合は、if "%destdir%"...の行を削除します。

destdir をセットしている箇所を if ~ else で表現するには次のようにします。

```
if "%2" == "" (
    set destdir=c:¥
) else (
    set destdir=%2
)
```