

Tips : スタンダード API を利用する

Copyright © EMIT Japan Corporation

概要

WebCT は Web による管理者インタフェースにより、ほとんどの設定を施すことが出来ます。しかし、それ以外にもいくつかのコマンドラインツールが用意されています。処理すべき項目が多いときなど、このツールを利用した方が作業量の軽減につながることもあります。

今回はコマンドラインツールの一つであるスタンダード API について紹介します。なお本稿は Linux 版でのみの記述になっておりますことを予めご了承下さい。

スタンダード API でできること

スタンダード API で可能な操作を以下の表にまとめます。

表 1 スタンダード API でできること

ユーザの追加	単一ないし複数のユーザ情報を追加する
ユーザ情報の更新	単一ないし複数のユーザ情報の更新。ユーザータイプを変更することも可能
ユーザの削除	単一ないし複数のユーザを削除する
ユーザ情報の取得	単一ないし複数のユーザ情報を取得する
WebCT ID の変更	単一ないし複数の WebCT ID を変更する

表 1 で見られるように、スタンダード API は主としてユーザの追加や削除などの操作をする目的で使用します。逆にコース情報の追加削除については行うことができません。コマンドラインによるコースの作成などは、IMS Enterprise AP¹をお使いいただくことになります。

¹ライセンスの種類によっては、お買い取り無効があります。Institution であれば可能です

この Tips は以下の内容を含んでいます。

- コマンドの場所
- スタンダード API チュートリアル
- スタンダード API コマンドリファレンス

コマンドの場所

スタンダード API のご利用について、インストールなどの作業は必要ありません。WebCT をインストールすると同時にスタンダード API も同時にお使いのシステムにインストールされます。インストールされる場所は以下のディレクトリになります。

```
$webct_root/webct/webct/generic/api/
```

ここで \$webct_root は webct をインストールしたディレクトリと読み換えてください。以降同じ表記を用いることにします。このディレクトリの下に webctdb がコマンドの実体になります。

スタンダード API チュートリアル

それではスタンダード API を用いたコマンドラインでの操作を行ってみましょう。簡易チュートリアルとして、スタンダード API を用いたグローバルデータベースへのユーザレコードの追加、コースへのユーザレコードの追加、またユーザレコードの検索および削除を行ないます。

準備

コマンドラインで操作するので WebCT が動作するファイルシステムにアクセスする必要があります。その他、ユーザデータベースを直接操作するため、必ず WebCT のバックアップを取っておく必要があります。WebCT のバックアップは WebCT を停止させた後、インストールディレクトリ全てを cp や tar 等で保存するだけで行うことができます。

```
$webct_root/webct/webctctl stop # webct 停止
cp -a $webct_root webct.old # コピー
$webct_root/webct/webctctl start # webct 再開
```

グローバルデータベースへの追加

さて、それではスタンダード API を利用してグローバルデータベースへ追加する作業を行ってみましょう。ここでは例として以下のユーザを登録することにします。

- WebCT ID: sato
- Password: 1234

以下のディレクトリにcd して下さい。

```
cd $webct_root/webct/webct/generic/api
```

cd できたら以下のコマンドを入力します。
(以下は複数行にわかれていますが、実際には一行で記述して下さい)

```
./webctdb add global xxxx  
"WebCT ID=sato,Password=1234" ", "
```

正常に追加できれば **Success:** と出力されて終了します。global の次の xxxx はそのまま入力してください。学生データベースを操作する際にはこの引数に追加情報を与えるのですが、グローバルデータベースの操作の場合は使用しないので、ダミーの引数を与える必要があるためです。(オフィシャルガイドであるテクニカルリファレンス²では xxxx を指定しています)

これでグローバルデータベースへの登録が完了しました。WebCT の URL をブラウザで開き、sato,1234 で正常にmyWebCT にログインできることを確認してください。

コースエントリの追加

さて、先の操作によりログインはできましたが、コースが一つも登録されていない為、「myWebCT にまだコースが追加されていません」と表示されます。先述したようにスタンダード API ではコースを追加することは出来ません。そこでアドミニストレータ インタフェースで **cs100** というコースを作っておく必要があります(デザイナーは適当なユーザ

を割りあてておきます)。コースが追加できたらコマンドラインに戻って、先程作成したユーザ sato を cs100 に学生として登録してみます。大文字小文字も区別しますから正しく入力してください。なおコンマの前後に空白を含んでもいけません。

```
./webctdb update global xxxx "WebCT ID=sato,  
Password=1234,Courses=cs100;S" ", "
```

正常に終了したら先程と同様にmyWebCT にアクセスしてみましょう。コース cs100 が追加されていれば成功です。

コースエントリの追加

続いて追加したコースを削除してみます。先程追加した cs100 を sato から削除します。

```
./webctdb delete student cs100 sato
```

グローバルデータベースから削除

最後に sato をグローバルデータベースから削除し、myWebCT にログインできない状態にしてみます。エントリを削除する場合は以下のコマンドを実行してください。この場合の xxxx もそのまま入力することになります。

```
./webctdb delete global xxxx sato
```

スタンダード API コマンドリファレンス

一通りグローバルデータベースの追加とコースエントリの追加を行ないました。以降、コマンドの使い方をリファレンス形式でまとめておきます。必要に応じて参照してください。

スタンダード API のコマンドラインインタフェースの実体は **webctdb** というプログラムです。この webctdb に第二引数を与えて、データベースの操作を決定することになります。webctdb と第二引数の組合せを以下の表にまとめました。

2

http://www.webct.com/support/viewpage?name=support_doc_index_campus_edition からダウンロードできますが ライセンスコンタクトパソンのメールアドレスとライセンスキーが必要です。

表 2 コマンド一覧

コマンド	動作
webctdb add	データベースに新規レコードを追加する
webctdb fileadd	ファイルから情報を読みこみ、データベースにレコードを一括して追加する
webctdb delete	データベースからエントリを削除する
webctdb filedelete	ファイルを読みこんでデータベースからエントリを一括して削除する
webctdb update	データベース情報を更新(上書き)する
webctdb fileupdate	ファイルを用いてデータベース情報を一括更新(上書き)する
webctdbchangeid	WebCT ID をファイルを用いて一括更新する (グローバルデータベースのみ)
webctdb filechangeid	WebCT ID を変更する (グローバルデータベースのみ)
webctdb find	WebCT データベースから指定した ID について、検索/情報取得する

以降これらのコマンドの詳細について記述します。<>で囲まれているものは必須オプション、[]で囲まれているものは任意のオプションになっています。

webctdb add

データベースに新規レコードを追加。

```
webctdb add <db> <course>
<fieldsData_pair_list>
<separator> [encrypted] [--charset=charset]
```

<db>

登録対象のデータベースを指定します。グローバルデータベースを指定する場合は **global**、学生データベースを指定する場合は **student** とします。

<course>

操作対象のコースを指定します。これは学生データベースを操作するときのみ必要になる引数ですが、グローバルデータベースを指定する場合はダミーの引数としてxxxxを与えてください。

<fieldsData_pair_list>

フィールドのリストを記述します。グローバルデータベースに含めるフィールドは以下になります。

- WebCT ID (必須)
- Password (必須)
- Courses
- First Name
- Last Name

```
"WebCT ID=student1"
```

上記はWebCT ID にstudent1 を指定した例です。上記の例のように、全体をダブルクォートで囲んで、キーと値は = で繋ぐ必要があります³。なお、大文字小文字は正しく記述し、前後に余分な空白を挟まないように注意してください。

後に指定する<separator>によって区切り、複数の値を設定することが可能です。以下の例を見てみましょう。

```
"User ID=ito,Password=1234,Last
Name=Ito,First Name=Masayo"
```

これは、(カンマ)によって区切った例です。実質的には以下を並べて指定していることになります。

- User ID=ito
- Password=1234
- Last Name=Ito
- First Name=Masayo

区切り文字は何でも良いのですが、<separator>のオプションで明示する必要があります。

グローバルデータベースのコースに権限付きで登録

³ シェルの制限です。

する場合は、コースと権限を;(コロン)および;(セミコロン)で区切る必要があります。そのためグローバルデータベース登録の際は、**セミコロンおよびコロンは区切り文字として使用する事ができない**ので注意してください。

以下の例を見てください。

```
"WebCT ID=sato,Password=1234,
Last Name=Sato,First Name=Shinichi,
Courses=cs100;S:cs200;TA"
```

先程と同じく、コンマ(,)で分解すると以下ようになります。

- WebCT ID=sato
- Password=1234
- Last Name=Sato
- First Name=Shinichi
- Courses=cs100;S:cs200;TA

Courses は、さらに :(コロン) によって分解される為

- CS100;S
- CS200;TA

になります。

さらに;(セミコロン)によって分割され

- CS100 S (学生)
- CS200 TA (ティーチングアシスタント)

となります。

指定出来る権限として「S(学生)、TA(ティーチングアシスタント)、D(デザイナー)」があります。

<separator>

<fieldsData_pair_list>の区切り文字を指定します。値として設定できるものとして英数字がありますが、:(コロン) 及び;(セミコロン)はグローバルデータベースにおいて<separator>として使用できないなどの制限があるので、一般的には(カンマ)を使うのが良いでしょう。

[encrypted]

passwordとして標準のUNIX DESを使用するときencryptedを明示的に指定します。add、update、fileadd、fileupdateにのみ使用可能です。

使用するときはencryptedと記述します。

[--charset=charset]

値として非アスキー文字を入れるときに指定する文字コードでデフォルトは管理者インタフェースで設定できる「APIインポートの文字コード」に依存します。"(ダブルクォート)で囲み--charset=文字コードという指定をします。お使いの端末の入力文字コードに合わせた指定にしてください。

以下の例を参照してください。

```
"--charset=euc-jp" # (EUC-JPを指定)
```

なお、指定できる文字コードの一覧については末尾の付録を参照して下さい。

Example

(以下のサンプルは複数行にわかれていますが、実際には一行で記述して下さい)

```
webctdb add global xxxx
"WebCT ID=kondo,Password=1111,
First Name=Isami,Last Name=Kondo" ", "
```

グローバルデータベースにWebCT Idがkondo、パスワード1111、姓がKondo、名がIsamiという人物を登録する。

```
webctdb add global xxxx
"WebCT ID=okita,Password=gQ3.OgePhgLuw,
First Name=Soushi,Last Name=Okita" ", "
encrypted
```

グローバルデータベースにWebCT IDがokita、パスワード2222、姓がOkita、名がSoushiという人物を、UNIX DESを用いたパスワードを使用して登録する。

```
webctdb add global xxxx
"WebCT ID=yamada,Password=yamada,
First Name=太郎,Last Name=山田" ", "
"--charset=shift_jis"
```

WebCT ID=yamada ,パスワード=yamada,姓=山田、名前=太郎 をソフト JIS 端末から登録する。

```
webctdb add global xxxx
  "WebCT ID=sato,Password=1234,
  Last Name=Sato, First Name=Shinichi,
  Courses=cs100;S:cs200;TA" ", "
```

WebCT ID=sato,パスワード1234,Shinichi Sato をコースcs100 には学生、cs200 にはティーチングアシスタントとして登録する。

(コースcs100 cs200 はあらかじめ存在していません。アドミン ストラータインタフェースで作成してください)

webctdb fileadd

ファイルから情報を読みこみ、データベースにレコードを一括して追加

```
webctdb fileadd <db> <course> <filename>
  <separator> [encrypt] [--charset=charset]
```

データベースにファイルからの情報を読みこみながら追加します。複数のユーザを追加するのに向いています。追加できるファイルの形式として以下を例にあげます。

```
WebCT ID,Password,Last Name,First Name
jsmith,1234,Smith,John
jbrown,2345,Brown,Jane
bfawlty,8765,Fawlty,Basil
arigsby,5432,Rigsby,Arthur
```

(このテキストはglobal.csv として保存し、後のexample で使うものとします)

また、Courses のフィールドを作成することも出来ます。この書式は webctdb add と同じく、コロンとセミコロンでコース名とユーザタイプを指定してください。

<db>

登録対象のデータベースを指定します。グローバルデータベースを指定する場合は **global**、学生データベースを指定する場合は **student** とします。

<course>

操作対象のコースを指定します。詳細は webctdb add の <course> の説明を参照下さい。

<filename>

作成したファイルのパスを指定します。

<separator>

ここでの <separator> 引数はファイルの区切り文字を指定します。上記の例のようにコンマ区切り(csv) のファイルを作成したのであれば "," のように指定します。

[encrypted]

password として標準の UNIX DES を使用するとき encrypted を明示的に指定します。add、update、fileadd、fileupdate にのみ使用可能です。使用するときには encrypted と記述するだけです。(詳細は webct add を参照してください)

[--charset=charset]

値として非アスキー文字を入れるときに指定する文字コードです。詳細は webctdb add の [--charset=charset] の説明をご覧ください。

Example

(以下のサンプルは複数行にわかれていますが、実際には一行で記述して下さい)

```
webctdb fileadd global xxxx global.csv ", "
```

先に示した global.csv をグローバルデータベースに追加する。

以下の日本語が含まれるファイルを user.csv としてカレントディレクトリに保存し、グローバルデータベースに追加する。

```
WebCT ID,Password,Last Name,First Name
hiromi,1234,佐藤,裕美
sinya,2345,岡本,真也
tatsuya,8765,柏木,達也
```

(日本語は EUC-JP で保存されているものとします)

```
webctdb fileadd global xxxx user.csv ", "
  "--charset=euc-jp"
```

webctdb delete

データベースからエントリを削除する

```
webctdb delete <db> <course>
<WebCT ID|user_id>
```

<db>

登録対象のデータベースを指定します。グローバルデータベースを指定する場合は **global**、学生データベースを指定する場合は **student** とします。

<course>

操作対象のコースを指定します。詳細は webctdb add の <course> の説明を参照下さい。

<WebCT ID|user_id>

操作対象の WebCT ID または User ID を指定します。グローバルデータベースのときは該当する WebCT ID、学生データベースのときは該当する User ID を指定して下さい。

Example

```
webctdb delete global xxxx sato
```

WebCT ID=sato をグローバルデータベースより削除する

```
webctdb delete student cs100 sato
```

User ID=sato をコースid=cs100 の学生データベースより削除する

webctdb filedelete

```
webctdb filedelete <db> <course> <filename>
```

ファイルを読みこんでデータベースからエントリを削除します。一括削除に向いています。ファイルの形式は以下のように WebCT ID ないし User ID が列挙されたものである必要があります。

```
s0001
s0002
s0003
s0004
s0005
```

(なお、ここで示したファイルは user.csv として後の Example で使用します。)

<db>

登録対象のデータベースを指定します。グローバルデータベースを指定する場合は **global**、学生データベースを指定する場合は **student** とします。

<course>

操作対象のコースを指定します。詳細は webctdb add の <course> の説明を参照下さい。

<filename>

作成したファイルのパスを指定します。

Example

```
webctdb filedelete global xxxx user.csv
```

user.csv という名前で保存された上記のファイルを用いてグローバルデータベースより削除する

```
webctdb filedelete student cs100 user.csv
```

user.csv という名前で保存された上記のファイルを用いて cs100 というコースの学生データベースより削除する

webctdb update

データベース情報を更新(上書き)する

```
webctdb update <db> <course>
<fieldsData_pair_list> <separator> [encrypt]
[--charset=charset]
```

データベース情報を更新します。概念としては上書きに近くなると思います。

基本的にこのコマンドは webctdb add と全く同じです。唯一違う点は、add が既に存在する ID (WebCT ID / User ID) に対してデータを上書き出来ないのに対し、update は既に存在する ID に対し上書き出来るという点です。

Example

(以下のサンプルは複数行にわかれていますが、実際には一行で記述して下さい)

```
webctdb update global xxxxx
"WebCT ID=sato,Password=2222,
First Name=Shinya,Last Name=Sato" ","
```

グローバルデータベースに存在するWebCT ID sato を、パスワード2222、姓をSato、名をShinyaに変更する。

webctdb fileupdate

ファイルを用いてデータベース情報を一括更新します。基本的にこのコマンドはwebctdb fileadd と全く同じです。唯一違う点はfileaddは既に存在するID(WebCT ID/user id)に対してデータを上書き出来ないのに対し、fileupdateは既に存在するIDに対し上書き出来るという点です。

詳細に関してはwebctdb fileadd もしくはwebctdb updateを参照下さい。

webctdb changeid

```
webctdb changeid global xxxxx
<fieldsData_pair_list> <separator>
```

WebCT IDを変更します。これは**グローバルデータベースのみ**に行なうことができる操作です。

<fieldsData_pair_list>

ここでの<fieldsData_pair_list>はwebctdb addのときに使用したものと異なり、旧IDと新IDのペアになります。以下のように設定します。

```
"Old ID=0001,New ID=0002"
```

<separator>

<fieldsData_pair_list>の区切り文字を指定します。

Example

(以下のサンプルは複数行にわかれていますが、実際には一行で記述して下さい)

```
webctdb changeid global xxxxx
"Old ID=sato,New ID=kato" ","
```

WebCT ID=sato をkatoに変更する

webctdb filechangeid

ファイルを用いてデータベース情報を一括更新(上書き)する

```
webctdb filechangeid global xxxxx
<filename> <separator>
```

WebCT IDをファイルを用いて一括変更します。これは**グローバルデータベースのみ**に行なうことができる操作です。複数のエントリのWebCT IDを変更するときに使用すると良いでしょう。ファイルとしては以下のようなものが使用できます。

```
Old ID,New ID
sato,kato
suzuki,suzumoto
ando,endo
```

(このファイルはchange id.csv として Example に使用します)

<filename>

使用するファイルのパスを指定する。

<separator>

区切り文字として指定する文字を指定する。

Example

(以下のサンプルは複数行にわかれていますが、実際には一行で記述して下さい)

```
webctdb filechangeid global xxxxx
changeid.csv ","
```

change id.csv で保存した上記のファイルを使用して、ID を一括変更する

webctdb find

WebCT Database から指定したIDについて、検索/情報取得する

```
webctdb find <db> <course>
<WebCT_ID|user_id> <separator> [user_type]
```

WebCT Database からIDを検索します。検索に指定できる条件はWebCT ID(グローバルデータベースのとき) User ID(学生データベースのとき)のみです。検索と書きましたが、

WebCT のクエリ検索のように、複数ユーザを検索することは出来ません。単一の ID を与えて、その ID のユーザの情報を取得するために利用します。

<db>

登録対象のデータベースを指定します。グローバルデータベースを指定する場合は **global**、学生データベースを指定する場合は **student** とします。

<course>

操作対象のコースを指定します。詳細は webctdb add の <course> の説明を参照下さい。

<WebCT_ID | user_id>

検索対象の WebCT ID (グローバルデータベースのとき)、もしくは User ID (学生データベースのとき) を指定します。

<separator>

ここでの <separator> 引数は入力ではなく、出力すべき区切り文字を指定します。

[user_type]

これを記述するとユーザタイプとして S, D, TA 付きの出力を返します。そのまま user_type と記述して下さい。詳細は Example をご覧下さい。

Example

(以下のサンプルは複数行にわかれていますが、実際には一行で記述して下さい)

```
webctdb find global xxxxx sato3 ", "
```

グローバルデータベースから WebCT ID が sato3 のものをカンマ区切りで出力する。

```
webctdb find global xxxxx sato3 "| "
```

グローバルデータベースから WebCT ID が sato3 のものを | 区切りで出力する。

```
webctdb find global xxxxx sato3 ", "  
user_type
```

グローバルデータベースから WebCT ID が sato3 のものをカンマ区切りで、ユーザタイプも併せて出力する。

```
webctdb find student cs100 sato3 ", "  
user_type
```

学生データベースから User ID が sato3 のものをカンマ区切りで、ユーザタイプも併せて出力する。

付録

--charset オプションに渡せる文字コード一覧

文字コード	オプション
中央ヨーロッパ言語 (ISO)	"--charset=iso-8859-2"
中央ヨーロッパ言語 (Windows)	"--charset=windows-1250"
キリル言語 (KO180R)	"--charset=ko18-r"
キリル言語 (Windows)	"--charset=windows-1251"
日本語 (EUC JP)	"--charset=euc-jp"
日本語 (Shift JIS)	"--charset=shift_jis"
簡体中国語	"--charset=gb2312"
繁体中国語	"--charset=big5"
トルコ語 (Windows)	"--charset=windows-1254"
ユニコード (UCS2)	"--charset=unicode2"
ユニコード (UTF-8)	"--charset=utf-8"
西ヨーロッパ言語 (ISO)	"--charset=iso-8859-1"

このTips は以下の環境で確認しました。

サーバ : WebCT4.0 日本語版 / RedhatLinux 7.3

クライアント OS : WindowsXP

クライアントブラウザ : IE6.0SP1 / NN7.1

(2004年4月14日 福山貴幸作成)