



## ACCELL/IDS から ACCELL/SQL へのコンバージョン

ACCELL/IDS インストールディレクトリ /home/AIDS  
ACCELL/SQL インストールディレクトリ /home/ASQL

### 1. SQL/Aデータベース設計変換スクリプトの作成

#### Step1 ACCELL/IDSの環境変数の設定

```
UNIFY=/home/AIDS/lib  
DBPATH=/home/ELS_DB  
DBNAME=file.db  
PATH=/home/AIDS:/home/AIDS/bin:$PATH  
REL2000=/home/ASQL  
CONV=/home/ASQL/conv
```

以下の2つはここでは必要ないが、ACCELL/IDSのアプリケーションを実行するときには必要になる。

```
TERMCAP=$UNIFY/termcap  
UNICAP=$UNIFY/unicap
```

#### Step2 ファイルのセーブ

DBPATH内のすべてのファイルをセーブし、DBPATHには3つのファイルのみ残しておく。

残しておいた3つのファイル

```
file.db  
file.dbr  
unify.db
```

#### Step3 DataServer用のデータベースを作成するためのディレクトリを作成し、ディレクトリを移る。

```
$ mkdir $DBPATH/../DS_DB  
$ cd $DBPATH/../DS_DB
```

#### Step4 変換準備ユーティリティの実行

```
$ $CONV/prep.sh
```

#### Step5 変換ユーティリティの実行

```
$ $CONV/conv.sh > conv.log 2>&1 &
```

#### Step6 prod.cfのコピー

\$UNIFYにあるprod.cfをfile.cfにコピーしSHMKEYは適当な数値を入れる。(例えば6904など)LMSHMKEYはコメントにしておく。

## 2. ACCELL/SQL(DataServer)データベースの作成

### Step1 ACCELL/SQL (DataServer)の環境変数の設定

```
UNIFY=/home/ASQL/lib
DBPATH=/home/DS_DB
DBNAME=file.db
PATH=/home/ASQL/bin:$PATH
```

以下の2つはここでは必要ないが、ACCELL/SQLのアプリケーションを実行するときには必要になる。

```
TERMCAP=$UNIFY/termcap
UNICAP=$UNIFY/unicap
```

### Step2 ACCELL/SQL (DataServer)のデータベースを作成する。

```
$ U2000.sh > sql.log 2>&1
```

### Step3 列グループの登録

```
$ Uaddcgp.sh
```

## 3. データ転送(DataServer ELS->DataServer)

### Step1 ACCELL/IDSの環境変数の設定

```
UNIFY=/home/AIDS/lib
DBPATH=/home/ELS_DB
DBNAME=file.db
PATH=/home/AIDS:/home/AIDS/bin:$PATH
```

### Step2 データのダンプ

```
$ cd /home/DB_DS/data
$ SQL ../U4.dump.sql > dump.log 2>&1 &
```

### Step3 ACCELL/SQL (DataServer)の環境変数の設定

```
UNIFY=/home/ASQL/lib
DBPATH=/home/DS_DB
DBNAME=file.db
PATH=/home/ASQL/bin:$PATH
```

### Step4 データのロード

```
$ sh ../U2000.load.dbld >dbld.log 2>&1 &
```

#### 4. ACCELLアプリケーションの変換

##### Step1 ACCELL/IDSの環境変数の設定

```
UNIFY=/home/AIDS/lib
DBPATH=/home/ELS_DB
DBNAME=file.db
PATH=/home/AIDS:/home/AIDS/bin:$PATH
TERMCAP=$UNIFY/termcap
UNICAP=$UNIFY/unicap
```

##### Step2 シェルを使ってフォームをASCIIフォーマットに変換

```
Q2ASC -a application_name.aq > application_name.az
for file in *.fq; do
    Q2ASC $file > `basename $file .fq`.fz
done
for file in *.hlp; do
    H2ASC $file > `basename $file .hlp`.hz
done
```

##### Step3 “= UNDEFINED”と”<> UNDEFINED”を”IS UNDEFINED”と”IS NOT UNDEFINED”に変換(= undefinedと<> undefinedについても同様)

```
for file in *.?s; do
    sed s/"= UNDEFINED"/"IS UNDEFINED"/g $file > $file.1
    sed s/"<> UNDEFINED"/"IS NOT UNDEFINED"/g $file.1 > $file.2
    mv $file $file.tmp
    mv $file.2 $file
    rm $file.1 $file.tmp
done
```

##### Step4 ACCELL/SQL (DataServer)の環境変数の設定

```
UNIFY=/home/ASQL/lib
DBPATH=/home/DS_DB
DBNAME=file.db
PATH=/home/ASQL/bin:$PATH
```

##### Step5 TRANを使ってACCELL/SQLフォーマットに変換

```
$ TRAN -S PUBLIC *z *s
```

##### Step6 z2q2zを使って、フォームへ変換

```
$ export CONV=/home/ASQL/conv
$ $CONV/z2q2z
```

##### Step7 makeファイルを作成し、コンパイル

```
$makeamake
$make
```

## 1. ACCELLユーザ関数(C-Hooks)の相違点

AVAL構造体に変更されたため、リターン値を定義する場合、avalmacs.hにあるマクロを使用することになった。それに伴い、ユーザ関数の変更が必要である。変更箇所は以下の通り。

旧コード (ACCELL/IDS)	新コード (ACCELL/SQL)
aval.dfflg = 0;	MKVALUDF(&aval);
aval->dfflg = 0;	MKVALUDF(avalp);
aval.dfflg = 1;	MKVALDFNN(&aval);
aval->dfflg = 1;	MKVALDFNN(avalp);
n/a	NULLを設定するには: MKVALDFNL(&avalp);
aval.dfflg == 1	ISVALDF(&aval)
aval->dfflg == 1	ISVALDF(aval)
aval.dfflg == 0	ISVALUDF(&aval)
aval->dfflg == 0	ISVALUDF(aval)

## 2. カスタムマネージャの再リンク

### Step1 ユーザ関数の変更

上記の対応表をもとに該当箇所をACCELL/SQLのコードに置き換える。

### Step2 ACCELL/SQL 環境でカスタムマネージャの再リンク

通常の方法でカスタムマネージャを再リンクする。ただし、ACCELL/SQLでアプリケーションのチェックを行わない場合には、このステップは省略できる。

例えば、namechk.cというファイルをコンパイルし、カスタムマネージャにリンクするには、以下のようにする。

```
$ cc -c -I$UNIFY/.. namechk.c
$ cc -c -I$UNIFY/.. chooktb.c

$ amgr.lid CUSTAMGR namechk.o chooktb.o
```