

目次

ACCELL/IDS , ACCELL/SQL 系

ACCELL/IDS で上位Releaseへのバージョンアップ	… 1
ACCELL/IDS から ACCELL/SQL へのコンバージョン	… 3
ACCELL/SQL Release1 から上位Releaseへのバージョンアップ	… 7
ACCELL/SQL Release2 以降で上位Releaseへのバージョンアップ	… 8

Unify Vision系

ACCELL/SQL から UnifyVISION Release2へのコンバージョン	… 9
ACCELL/SQL から UnifyVISION Release2へのコンバージョン(C-Hooks)	… 12
ACCELL/SQL からUnifyVISION Release3以降のリリースへのコンバージョン	… 13



ACCELL/IDS で上位Releaseへのバージョンアップ

ACCELL/IDS 現リリースインストールディレクトリ /home/AIDS
ACCELL/IDS 新リリースインストールディレクトリ /home/AIDS_NEW

[注釈]

ACCELL/IDS(UNIFY4.0対応)でACCELL/IDS(UNIFY DataServerELS対応)にバージョンアップを行なう場合は、データベースの移行作業（UNIFY4.0からUNIFY DataServerELS）を行なった後に、下記の作業を行なう。

[注釈]

ACCELL/IDS(UNIFY DataServerELS対応)で上位リリースにバージョンアップを行なう場合は、データベースの移行作業は不要。

1. ACCELLアプリケーションの変換 Part.1(ACCELL/IDS 現リリース)

Step1 ACCELL/IDS 現リリースの環境変数の設定

```
DBPATH=/home/ELS_DB ; export DBPATH
PATH=/home/AIDS:/home/AIDS/bin:$PATH ; export PATH
UNICAP=/home/AIDS/lib/unicap ; export UNICAP
TERMCAP=/home/AIDS/lib/termcap ; export TERMCAP
UNIFY=/home/AIDS/lib ; export UNIFY
```

Step2 フォームをASCIIフォーマットに変換

```
マスターフォームの場合、 Q2ASC -a フォーム名 > フォーム名.az
一般フォームの場合、    Q2ASC フォーム名 > フォーム名.fz
ヘルプフォームの場合、  H2ASC フォーム名 > フォーム名.hz
```

2. ACCELLアプリケーションの変換 Part.2(ACCELL/IDS 新リリース)

Step1 ACCELL/IDS 新リリースの環境変数の設定

```
DBPATH=/home/ELS_DB ; export DBPATH
PATH=/home/AIDS_NEW:/home/AIDS_NEW/bin:$PATH ; export PATH
UNICAP=/home/AIDS_NEW/lib/unicap ; export UNICAP
TERMCAP=/home/AIDS_NEW/lib/termcap ; export TERMCAP
UNIFY=/home/AIDS_NEW/lib ; export UNIFY
```

Step2 アスキー化したフォームファイルから、ACCELL/IDS 新リリース対応のフォームファイルを作成する。

```
マスターフォームの場合、 ASC2Q -a フォーム名 < フォーム名.az
一般フォームの場合、    ASC2Q フォーム名 < フォーム名.fz
ヘルプフォームの場合、  ASC2H フォーム名 < フォーム名.hz
```

Step3 makeファイルを作成し、コンパイル/結合/リンク

```
$makeamake
$make
```

Step4 システム・タイトルの変更

シェルスクリプト `acell` を実行してメインメニューを立ちあげた際、画面に表示されるシステム表題が、前のリリースのバージョンである場合は、「システムパラメータの変更 (parmnt)」を使用して、システム表題を新しいリリースのバージョンに併せて変更する。

"ACCELL Release 1.4E"

↓ 変更

"ACCELL Release 8.1H"



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

```
$ Uaddecgp.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構造体が変更されたため、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.ld CUSTAMGR namechk.o chooktb.o
```



ACCELL/SQL Release1から上位Releaseへのバージョンアップ

ACCELL/SQL Release1インストールディレクトリ	/home/ASQL1
ACCELL/SQL 新リリースインストールディレクトリ	/home/ASQL_NEW

現在のACCELL/SQL対応データベースがUNIFY2000で、ACCELL/SQL 新リリースの対応データベースがDataServerとなる場合は、データベース移行処理 (UNIFY2000 → DataServer)を行なった後に、下記の作業を行なう。

1. ACCELLアプリケーションの変換Part.1(ACCELL/SQL Release1)

Step1 ACCELL/SQL Release1の環境変数の設定

```
UNIFY=/home/ASQL1/lib
DBPATH=/home/DB_DS
DBNAME=file.db
PATH=/home/ASQL1/bin:$PATH
TERMCAP=$UNIFY/termcap
UNICAP=$UNIFY/unicap
```

Step2 フォームをASCIIフォーマットに変換

```
マスターフォームの場合、 Q2ASC -a フォーム名 > フォーム名.az
一般フォームの場合、    Q2ASC フォーム名 > フォーム名.fz
ヘルプフォームの場合、  H2ASC フォーム名 > フォーム名.hz
```

2. ACCELLアプリケーションの変換Part.2(ACCELL/SQL)

Step1 ACCELL/SQL 新リリースの環境変数の設定

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

Step2 TRANを使って、フォームファイルと4GLスクリプトファイルを
ACCELL/SQL フォーマットに変換

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

Step3 アスキー化したフォームファイルから、ACCELL/SQL 新リリース用のフォーム
ファイルを作成する。

```
マスターフォームの場合、 ASC2Q -a フォーム名 < フォーム名.az
一般フォームの場合、    ASC2Q フォーム名 < フォーム名.fz
ヘルプフォームの場合、  ASC2H フォーム名 < フォーム名.hz
```

Step4 makeファイルを作成し、コンパイル/結合/リンク

```
$makeamake
$make
```



ACCELL/SQL Release2 以降で上位Releaseへのバージョンアップ

ACCELL/SQL 現インストールディレクトリ	/home/ASQL
ACCELL/SQL 新リリースインストールディレクトリ	/home/ASQL_NEW

現在のACCELL/SQL対応データベースがUNIFY2000で、ACCELL/SQL 新リリースの対応データベースがDataServerとなる場合は、データベース移行処理 (UNIFY2000 → DataServer)を行なった後に、下記の作業を行なう。

1. ACCELLアプリケーションの変換Part.1(ACCELL/SQL 現リリース)

Step1 現在のACCELL/SQL の環境変数の設定

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

Step2 フォームをASCIIフォーマットに変換

```
マスターフォームの場合、 Q2ASC -a フォーム名 > フォーム名.az
一般フォームの場合、    Q2ASC フォーム名 > フォーム名.fz
ヘルプフォームの場合、  H2ASC フォーム名 > フォーム名.hz
```

2. ACCELLアプリケーションの変換Part.2(ACCELL/SQL 新リリース)

Step1 ACCELL/SQL 新リリースの環境変数の設定

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

Step2 アスキー化したフォームファイルから、ACCELL/SQL 新リリース用のフォームファイルを作成する。

```
マスターフォームの場合、 ASC2Q -a フォーム名 < フォーム名.az
一般フォームの場合、    ASC2Q フォーム名 < フォーム名.fz
ヘルプフォームの場合、  ASC2H フォーム名 < フォーム名.hz
```

Step3 makeファイルを作成し、コンパイル/結合/リンク

```
$makeamake
$make
```



ACCELL/SQL からUnifyVISION Release2へのコンバージョン

ACCELL/SQL インストールディレクトリ

/home/ASQL

UnifyVISION Release2インストールディレクトリ

/home/VISION2

1. UnifyVISIONアプリケーションの変換

Step1 UnifyVISIONとDataServer の環境変数の設定

```
PATH=/home/VISION2/bin:$PATH
VISION_HOME=/home/VISION2
GALAXYHOME=/home/VISION2/gui
LANG=japanese
LANG_DIR=jpn_jae
DISPLAY=jupiter
UNIFY=/home/ASQL/lib
CLIENTINFO=/home/DS_DB
DCMFILE=/export/home/ohmura/vdev/tutorial/tutorial.dcm
DBCONN=TUTORIAL._U2K
```

【tutorial.dcmの内容】

```
[TUTORIAL_U2K]
DBTYPE=U2000
DBHOST=jupiter
DBUSER=ohmura
DBPASSWORD=BeEsXRO
DBNAME=file.db
DBPATH=/home/DS_DB
DBSCHEMA=PUBLIC
```

注) これらの設定は、使用する環境により変更する。

注) *.az, *.fzファイルがない場合にはStep1の前に作成しておく。

Step2 makefileの作成と編集

ACCELLアプリケーションのあるディレクトリに移り、acl2uvユーティリティを実行する。

```
$ acl2uv
```

makefileのvcpl部分を以下のように変更する。

【変更前】

```
-vcpl -warm all -log $*.log -script $?
```

【変更後】

```
-vcpl -warm all -log $*.log -script $? -dcmfile $(DCMFILE)
      -dbconn $(DBCONN)
```

Step3 makeの実行

```
$ make -f vision.mak
```

Step4 visionを起動しStep3で作成したアプリケーションを取り込む。

プロファイルのデータベース・プリファレンスを設定して、接続を確認する。

Step5 Step3で作成されたログ(filename.log)を参照して4GLの編集を行う。
 以下の変更なしでもアプリケーションのコンパイルは問題なくできるが、将来にわたってUnifyVISIONを使用することを前提に変更すべき箇所を以下に示す。
 ここに示したものは、使用頻度が高いと思われるもの。

UnifyVISIONで変更された4GL文(左側:ACCELL 右側:UnifyVISION)

●APPLCAITION master_form	FORM CLASS master_form
●AFTER APPLICATION	ON DESTROY
●BEFORE FORM	ON CREATE
◎ENABLE ZOOM TO	ENABLE ZOOM TO CHILD FORM form_name OF CLASS form_name
●CHOOSE NEXT FORM	ON CHOOSE NEXT FORM
●ON PREVIOUS FORM	ON DISMISS FORM

●PREVIOUS_FORM	DISMISS_FORM

●NEXT_FIELD	NEXT_FIELD_NAME
●CUR_NEXT_FIELD	CURRENT_NEXT_FIELD_NAME

UnifyVISIONで追加されたもの

◎FIELD SECTIONにBEGIN ENDを入れる

UnifyVISIONで削除されたもの

- ◎DISPLAY TRIM
- ◎REFRESH SCREEN
- ◎REQUIRED FORMS
-
- ◎BLINK
- ◎LOW_INTENSITY
- ◎REVERSE
- ◎UNDERLINE

UnifyVISIONで設定が細かくなったもの

AUTO_COMMITについては、追加・削除・更新などについて別々に指定できるようになった。

◎ AUTO_COMMIT TX_MODE_ADD_RECORD
TX_MODE_DELETE_RECORD
TX_MODE_UPDATE_RECORD

SET COMMANDはなくなり、アトリビュート(属性)で指定するようになった。例えば、

◎SET COMMAND ‘NEXT_FORM’:ACTION TO ‘DISABLED’

は、以下のように変える必要がある。

SET ‘NEXT_FORM’:AUD_ACTION TO ‘DISABLED’

また、ACTIONについてもAUD_ACTIONとFIND_ACTIONの2つを指定する必要がある。

UnifyVISIONで条件が厳しくなったもの

FINDなど、QUEUE COMMANDに置き換えが必要なものがある。

◎NEXT ACTION IS QUEUE COMMAND

以上の変更は、ac12uvを実行する前にシェルスクリプトなどで行っても良い。ただし、ログにはACCELLでは未知のキーワードであるとのメッセージが出力される。

NULLの設定は、NULLを指定して関数を使って変換する。例えば、DATEタイプでは、以下のように行う。

◎SET TODAY_DATE TO str_to_date\$('*/*/*')

は、以下のように変更する。

SET TODAY_DATE TO to_date\$(NULL)

UnifyVISIONデザイナーで修正が必要なもの

◎リターンを押しても次フィールドにいかない

各フィールドの“タブストップ”属性をTRUEに設定する。

(フォーム・プロパティパネルのフィールド順で“タブストップ”をクリックする)

◎マウスでクリックしてフィールド間を移動できない

フォーム属性のフィールド・クリックをTRUEに設定する。

(フォーム・プロパティパネルの対話型操作で“フィールド・クリック”をクリックする)



ACCELL/SQL からUnifyVISION Release2へのコンバージョン(C-Hooks)

ACCELL/SQL インストールディレクトリ /home/ASQL
UnifyVISION Release2インストールディレクトリ /home/VISION2

1. UnifyVISIONユーザ関数(C-Hooks)の相違点

ACCELL/IDSからのコンバージョンの場合には、“ACCELL/IDSからACCELL/SQL へのコンバージョン(C-Hooks)”の“1. ACCELLユーザ関数(C-Hooks)の相違点”を参照のこと。ACCELL/SQLとUnifyVISIONは同一仕様。

ただし、インクルードしている`chookincl.h`を`chookinc.h`に変更すること。

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 (VISION)のコードに置き換える。

Step2 UnifyVISION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。

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。リンク前には`HDATYPE`を設定する必要がある。`PATH`の設定やデータベース固有の設定(例えば`ORACLE_HOME`など)もしておく必要がある。

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

$ HDATYPE="U2000 ORACLE"; export HDATYPE
$ vision.ld custvisn namechk.o chooktb.o
```

メモリを大量に消費するため、失敗した場合には、以下のように`swap`を追加して再度行う。

```
Solaris2.xの例
# mkfile 30M /export/home0/swap
# swap -a /export/home0/swap
```



ACCELL/SQL からUnifyVISION Release3以降のリリースへのコンバージョン

ACCELL/SQLインストールディレクトリ /home/ASQL
UnifyVISION リリースインストールディレクトリ c:\¥vision

ACCELL/SQL からUnifyVISION Release3以降のリリースへのコンバージョンは、1度に行うことができない。

ACCELL/SQL → UnifyVISION Release2 → UnifyVISION 新リリースと2段階のコンバージョン作業を行う必要がある。

1. ACCELL/SQL からUnify VISION Release2へのコンバージョン

コンバージョン作業の詳細は、「ACCELL/SQL から UnifyVISION Release2へのコンバージョン」に関するドキュメントを参照する。

2. Unify VISION Release2からRelease3以降の上位リリースへのバージョンアップ

コンバージョン作業の詳細は、「UnifyVISION Release2からRelease3以降の上位リリースへのバージョンアップ」に関するドキュメントを参照する。