
Exporting DataFax data into Relational Databases for Reporting and Analysis

Kerrie Roeber - Principal IT Developer

Arrhythmia Management Clinical & Outcomes Research Department

James Johnson - Sr Mgr, Clinical IS and Biostatistics

Cardiac Resynchronization Department

Exporting DataFax data into Relational Databases for Reporting and Analysis

- Why are we doing this
- Planning the end result
- Stepping through the code
- Validating the extraction
- Sample reports created with data extracted
- Questions

Purpose

- Create reports using Business Objects®
- Analysis using SAS®
- Data transfer to study management system

Things to consider

- How will data be merged
 - 1 plate per table or many plates per table
- Duplicate variables names
- How often will the tables be refreshed

Our System

- DataFax 3.6 runs on Sun Enterprise server with Sun Solaris 8 for SPARC OS
- Oracle ® 8.1.7.4.0 runs on a Sun Fire 420R with Sun ® Solaris 8 for SPARC OS
- SAS ® runs on a Sun ® Ultra 10 with Solaris 8, to meet SAS ® license agreements.

Step 1 – Create a script to create the DFsas job file

```
echo ''
echo 'createsas: creates DFsas files for each form, e.g. implant.sas'
echo ''
cd /home1/studies/SMS_Test/DFUG
*This is the directory where the DFsas files are created
echo 'Creating DFsas job on DFUG data . .'
echo ''
rm dfug
*Removes the existing file for the DFUG CRF
DFsas dfug -C 018 -p 300-301 -d s
```

Step 1 Continued

DFsas dfug -C 018 -p 300-301 -d s

dfug is the name of the file being created

-C includes all variables

-c includes all variables except

- *DFRASTER*
- *DFSTUDY*
- *DFPLATE*
- *DFCREATE*
- *DFMODIFY*
- *DFSCREEN*

Step 1 Continued

DFsas dfug -C 018 -p 300-301 -d s

018 is the study number

-p 300-301 includes plate number 300-301

-d s exports dates with a string format

Can also add c – calendar, o - original and j - julian format

- See DataFax Programmer Guide Chap 10 DFsas: DataFax to SAS for more formatting options

DFsas File created

```
# CREATED BY: amcis Jan 20,2004 10:10AM using
DFsas
# VERSIONS: DFsas 3.6, 03/03/04 and
.DFsas.awk 3.6, 03/03/04
# GLOBAL SPECIFICATIONS
DFNUM 018
SASDIR /home1/studies/SMS_Test/DFUG
SASJOB dfug
SASVERSION none
PATIENTS all
VISITS all
OPTIONS YEARCUTOFF=1920
IMPUTE no
INFORMAT dates
IDNAME PTID
CHECK labels
CHOICE codes
VLABEL yes
VALFMT yes
BLANK all asis
MISSING all asis
NOVISIT .
RECSTATUS clean dirty
MERGE yes
```

```
# DATA RETRIEVALS

RECORD 300
1 DFSTATUS DataFax Record Status
2 DFVALID DataFax Validation Level
6 DFSEQ DataFax Sequence Number
8 PTIDENT 1.1 Patient ID
9:s CONSDT 1.2 Consent Date
10:s BDT Birth date
11 GENDER Gender
12 CHECKBOX Checkbox

RECORD 301
1 DFSTATUS DataFax Record Status
2 DFVALID DataFax Validation Level
6 DFSEQ DataFax Sequence Number
8 PERSON 11.1 Person complet form
9 INVEST 11.2 Invest name
10 INVSIGN 11.3 Investigator sign
11:s AUTHDT 11.4 Date of auth sign
```

Step 2 – Create a script to update and execute the DFsas job file.

```
echo 'Executing DFsas job on DFUG data . . '
```

```
echo ''
```

```
sed -e 's/CHECK labels/CHECK codes/' dfug > dfug.tmp
```

Replaces the CHECK labels with CHECK codes. It will export the number associated with check boxes instead of the text "Checked" or "Not Checked" in a temporary file

```
mv dfug.tmp dfug
```

Overwrites the original file with the temporary file

```
DFsas dfug -f
```

Executes the DFsas file and forces it to create an output data file even if there is no data

- Executing this script will create a SAS job file and data files, 1 per plate

Step 2 Continued

DFsas dfug -f

Executes the DFsas file and forces it to create an output data file even if there is no data

- Executing this script will create a SAS job file and data files, 1 per plate

Sample SAS File

```

/* CREATED BY: amcis Jan 20,2004 10:10AM using DFsas */
/* VERSIONS: DFsas 3.6, 03/03/04 and .DFsas.awk 3.6, 03/03/04 */
options YEARCUTOFF=1920;

proc format ;
  value F0001v 9999 = "Not Checked" ;
  value F0002v 9999 = "Not Checked"
    1 = "Checked" ;

filename data1
  /net/maktub.pace/home1/studies/SMS_Test/DFUG/dfug.d01;
data data1 (label="DFUG Presentation 1");
infile data1 LRECL=4096 dlm='|' missover dsd;
informat PTIDENT $CHAR20. ;
informat CONSDT $CHAR11. ;
informat BDT $CHAR11. ;
input PTID DFSTATUS DFVALID DFSEQ PTIDENT $ CONSDT $ BDT
  $
  GENDER CHECKBOX ;
format DFSTATUS F0001v. ;
format GENDER F0001v. ;
format CHECKBOX F0002v. ;
label PTID="ID number"
  DFSTATUS="DataFax Record Status"
  DFVALID="DataFax Validation Level"
  DFSEQ="DataFax Sequence Number"
  PTIDENT="1.1 Patient ID"
  CONSDT="1.2 Consent Date"
  BDT="Birth date"
  GENDER="Gender"
  CHECKBOX="Checkbox";

filename data2
  /net/maktub.pace/home1/studies/SMS_Test/DFUG/dfug.d02;
data data2 (label="DFUG Presentation 2");
infile data2 LRECL=4096 dlm='|' missover dsd;
informat PERSON $CHAR75. ;
informat INVEST $CHAR75. ;
informat AUTHDT $CHAR11. ;
input PTID DFSTATUS DFVALID DFSEQ PERSON $ INVEST $
  INVSIGN
  AUTHDT $ ;
format DFSTATUS F0001v. ;
format INVSIGN F0002v. ;
label PTID="ID number"
  DFSTATUS="DataFax Record Status"
  DFVALID="DataFax Validation Level"
  DFSEQ="DataFax Sequence Number"
  PERSON="11.1 Person complet form"
  INVEST="11.2 Invest name"
  INVSIGN="11.3 Investigator sign"
  AUTHDT="11.4 Date of auth sign";

data final;
merge data1 data2;
by PTID;

```

Sample data file

- 999999001|1|1|300|JANE
DOE|01/JAN/1960|01/JAN/1960|01/JAN/1954|01/JAN/1954|1|1
- 999999001|1|1|300|JON DOE|DR
SMITH|1|14/JAN/2004|14/JAN/2004

Step 3 – Create SAS files to load data into Oracle tables

```
filename Copylib '/ home1/studies/SMS_Test/DFUG ';  
%include Copylib(dfug);
```

```
proc sort data=data1;  
  by PT FORMID;  
run;
```

```
proc sort data=data2;  
  by PT FORMID;  
run;
```

```
data final;  
  merge data1 data2;  
  by PT FORMID;
```

Merges data from multiple plates into 1 table and sorts by patient id and the formid.

Step 3 Continued

* Create ORACLE tables.

*,

```
%include "/home1/studies/SMS_Test/oracle/oraclepath.sas";
```

The Oracle login information is stored in a separate file

```
proc sql;
```

```
drop table oraam.dfug;
```

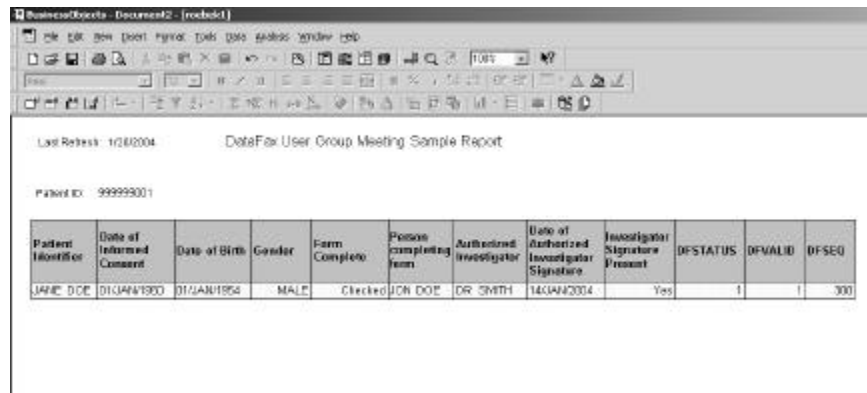
```
create table oraam.dfug as select * from final;
```

```
quit;
```

Validation

- DF lite is used to print the data entered into DataFax
- DF lite print out is compared to what is populated into the Oracle table.

Sample Business Objects Report



The screenshot shows a Business Objects report viewer window titled "BusinessObjects - Document2 - (rochdel)". The report content includes:

Last Refresh: 1/20/2004 DataFox User Group Meeting Sample Report

Patient ID: 999993001

Patient Identifier	Date of Informed Consent	Date of Birth	Gender	Form Complete	Person completing form	Authorized Investigator	Date of Authorized Investigator Signature	Investigator Signature Present	DFSTATUS	DFVALID	DFSEQ
JANE DOE	01/JAN/2000	01/JAN/1954	MALE	Checked	JON DOE	DR SMITH	14/JAN/2004	Yes	1	1	300

Questions