Programming DataFax

Eric Bosch
Clinical DataFax Systems Inc

DataFax Programming Tools

- Edit checks
  Interactive and batch
- Command-line programs
  `DFimport.rpc`, `DFexport.rpc`, `DFgetparam.rpc`, `DFlistplates.rpc`, etc.
- Reports
- `DFsas`
- Shell scripts
  Triggered by events, user request, scheduling
  Really the glue that holds the pieces together

Shell Scripting

- Helpful for gluing together existing pieces to build a solution
- Well-written scripts become useful pieces for building other solutions
- Typically Bourne shell syntax
- Easy to learn and create incrementally

Shell Scripting - Recommendations

- Be explicit with pathing to fixed components such as commands, or use `$PATH` variable
  e.g. `$DATAFAX_DIR/bin/DFlistplates.rpc`
- Use `DFgetparam.rpc` to locate study-specific information
- Parse arguments with `getopts`
- Check exit status of interim commands and abort cleanly
- Exit with status of 0 if no errors
  This is being a good neighbor so that other scripts can use your script.

Plate Arrival Actions

- User-specified script/program to execute whenever a particular plate number arrives
- Program can execute before or after ICR is performed (or both)
- DataFax makes `$image` (CRF image name) and `$data` (ICRed data record) variables available to program

- Pros:
  - Automated and quick - actions occur as soon as CRF arrives to study
- Cons:
  - Delayed if CRF bacode is not readable or study is disabled/read-only
  - ICRed data has not been verified

Plate Arrival Actions

- Typical example is to notify someone when an adverse event CRF arrives

Print the CRF

```
DATAFAX_DIR=/opt/datafax; export DATAFAX_DIR; \
$DATAFAX_DIR/lib/DFprint_filter $image*
```

Notify via email

```
echo "An AE has arrived" | mail username@emailaddress
```

Forward the CRF via email

```
DATAFAX_DIR=/opt/datafax; PATH="$PATH:$DATAFAX_DIR/bin"; \
export DATAFAX_DIR PATH; cd /tmp; DFpsprint $image* | DFgs \
-sOutputFile=ae.pdf -sDEVICE=pdfwrite -q -; DFsendfax \
ae.pdf mailto:username@emailaddress; rm ae.pdf
```

```
* DFimageio `echo $data | awk -F\| '{ print $3 }'`
```

Edit Checks

- Program for interactive users
  Preferable when user is needed for feedback or final decision
- Program for batch users
  Preferable when cross-plate and cross-visit checks are needed
- Extensible with dfrun() and dfsystem()
  If the language doesn't offer something, create a shell script to do it
- Be careful with processes that are expected to occur only once
  Field enter/exit can be traversed more than once
  Use plate enter/exit and check DFSTATUS==0 || DFVALID==0

Creating a New Report

- Most reports are just shell scripts!
- Must have execute permissions
- The first argument, $1, is always the study number when run from DFreports
  Good idea to standardize for all reports
- Installed in $DATAFAX_DIR/reports for general use
  Documented in $DATAFAX_DIR/reports/.info
- Installed in $STUDY_DIR/reports for study-specific use
  Documented in $STUDY_DIR/reports/.info
- Secret benefit of running reports from the command-line or cron (P.S. don't tell Wayne):
  No license is used!

Customizing DataFax Reports

- Edit the standard script to provide customized behavior/output
  - Recommend naming it uniquely so that it is immune to future releases
- Post-process the standard output
  - Example: Ignore complaints about missing 0000/9999999 images from DF_ICimages output

```
#!/bin/sh
$DATAFAX_DIR/reports/DF_ICimages "$@" | sed -e \
'/^0000\/9999999/d'
exit $?
```

Combining DataFax Reports

- Example: Create a new report that generates and sends a current QC report for specified center(s)
- Run DF_QCupdate, followed by DF_QCreports, and then DF_QCfax
- Parse arguments to determine requested center number(s), if any
- Call it `MY_QCreporter`

Combining DataFax Reports (2)

```
#!/bin/sh
USAGE()
{
    echo "MY_QCreporter [-u] [-c #]"
}
. $DATAFAX_DIR/reports/.report_setup
DFNUM=$1
shift

centers=""
while getopts uc: arg
do
    case $arg in
        c) centers="-c $OPTARG";;
        u) USAGE(); exit 0;;
        \?) USAGE(); exit 1;;
    esac
done
```

Combining DataFax Reports (3)

```
$DATAFAX_DIR/reports/DF_QCupdate $DFNUM
if [ $? -ne 0 ]; then
    echo "DF_QCupdate failed"
    exit 1
fi
$DATAFAX_DIR/reports/DF_QCreports $DFNUM $centers
if [ $? -ne 0 ]; then
    echo "DF_QCreports failed"
    exit 1
fi
$DATAFAX_DIR/reports/DF_QCfax $DFNUM
exit $?
```

DFsas Without SAS

- DFsas creates input data files for SAS
  But the input data files don't have to go to SAS (only)
- DFsas has capabilities (without SAS) for
  - re-coding data values
    - `BLANK`, `RECODE`, `MISSING` statements
    - `CHECK`, `CHOICE`, `NUMBER`, `STRING` statements
  - filtering by data values
    ```
    if ( sex == 1 && age > 65 )
    ```
  - normalization
    ```
    NORMALIZE
    ```

  These capabilities are not readily available in other DataFax programs

"More than one way to peel a potato"

Example: create a listing of each patient id and their age
Solutions:

1. Unformatted: export required fields from plate record
   ```
   DFexport.rpc -s primary -G "ID,AGE" 254 1 -
   ```
2. Semi-formatted: `DF_PTlist`
   ```
   DF_PTlist 154 [-d] -x 1 9 "%10.10s"
   ```
3. Complete formatting control: export to Excel
   ```
   DFexport.rpc -s primary -z -G "ID,AGE" 254 1 -
   ```
4. DFsas and create listing in SAS
5. DF_stats to get statistics about age
   ```
   DF_stats 254 -P 1 -F 9
   ```

Pulling it together

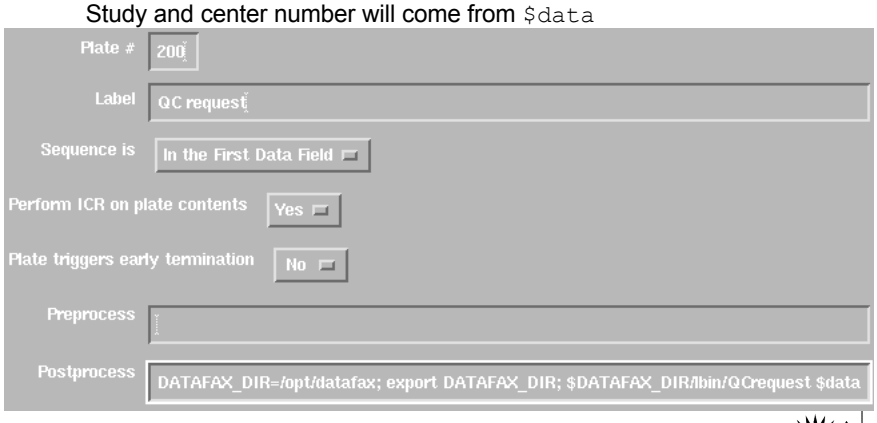
- Example: Site-driven QC reports
- Triggered by arrival from the site of a special plate
- Upon arrival of plate, QC report is created and sent back to site

Your center number: 0030

Fax this form to 888-555-1212 to request your QC report.

Pulling it together (2)

- Define new plate in DFsetup
  - Important that visit number not be in visitmap range
  - Study and center number will come from `$data`



Pulling it together (3)

- `QCrequest` script:
  Installed in `$DATAFAX_DIR/lbin` with execute (555) permissions

```
#!/bin/sh
PATH="/bin:/usr/bin:$DATAFAX_DIR/bin"; export PATH
USAGE()
{
    echo "QCrequest indata"
}

if [ $? -ne 1 ]; then
    USAGE | DFlogger -t QCrequest
    exit 1
fi
IN=$1
DFNUM=`echo '$IN' | DFget 4`
CTRNUM=`echo '$IN' | DFget 7`
```

Pulling it together (4)

```
# Do we have a center number?
if [ $CTRNUM="" -o $CTRNUM -eq 0 ]; then
    echo "Received '$IN' without center#" | DFlogger -t QCrequest
    exit 1
fi

# Our previously created script/report does the work
$DATAFAX_DIR/reports/MY_QCreporter $DFNUM -c $CTRNUM
exit $?
```

Pulling it together (5)

- Pros
  - Puts site in control of QC report scheduling
  - Leads to increased "ownership" at site
- Cons
  - Delivers QC report to all QC recipients for site, not just requester
  - Plate 200 records end up in new queue and need to be deleted
  - Will fail if barcode or center number not read (correctly) - this is easily remedied by providing a plate 200 PDF for sites to email

Further Information

- DataFax Documentation

  Standard Reports Guide

  `http://www.datafax.com/Support/doc/reportsman/html/index.html`

  Programmer Guide

  `http://www.datafax.com/Support/doc/progman/html/index.html`
- DataFax Tips of the Week

  `http://www.datafax.com/Support/tips/20010917.html`

  `http://www.datafax.com/Support/tips/20001211.html`

  etc.
- Shell Scripting

  Google for "unix shell scripting books"