

---

# DataFax Power Tools

Tools of the Trade

Darryl Pahl  
DF/Net Research, Inc.



## What are Power Tools?

---

- Simple tools, scripts, programs, tips, and techniques
- Designed with a minimal amount of fuss *just to get the job done*

So....

- They might not work for you
- They're not complete applications
- Be sure to wear safety glasses at all times



## How Good Is Your Data Entry?

---

- Two passes of data entry common in DataFax (1<sup>st</sup> + 2<sup>nd</sup> pass review)
- How can we measure the accuracy of your 1<sup>st</sup> pass?
- Here's a way to re-tool the standard DataFax Double Data Entry (DDE) tools to provide the answer...



## DDE vs. Enter and Review

---

- Classic DDE compares 1<sup>st</sup> entry to 2<sup>nd</sup> entry using "DDE sets" and data retrieval files (.drfs)
- In our case, we want to mimic DDE while still following 1<sup>st</sup> entry / 2<sup>nd</sup> review procedures
- Compares the 1<sup>st</sup> entry to the last version of the 2<sup>nd</sup> review



## DF\_compare

---

- A perl program that uses *standard* DataFax DDE tools to compare 1<sup>st</sup> and 2<sup>nd</sup> pass data entry
- Very useful as a training and monitoring tool for data entry staff
- Can provide excellent CRF design metrics
- Could easily be modified to include random 3<sup>rd</sup> pass sampling



## Output of DF\_compare

---

JNL,DDE	#RECS	INTEGER		DATES		CHOICES		CHECKS		STRINGS		TOTAL	
		Dis	%Dis	Dis	%Dis	Dis	%Dis	Dis	%Dis	Dis	%Dis	Dis	%Dis
kell,2nd	427	4	0.1	0	0	5	0.1	9	0.4			18	0.2
lori,2nd	115	0	0	1	0.3	1	0.1	1	0.2			3	0.1
jerr,2nd	1334	23	0.2	17	0.6	3	0	2	0			45	0.1
bill,2nd	1430	15	0.1	22	0.7	21	0.2	2	0			60	0.2
tery,2nd	994	47	0.5	5	0.2	14	0.2	23	0.5			89	0.4
dick,2nd	66	1	0.2	0	0	0	0	0	0			1	0.1
harr,2nd	4114	91	0.2	30	0.3	13	0	31	0.2			165	0.2



## Output of DFdde.rep

```
DFdde.rep 20 -set first_pass
DDE DISCREPANCY REPORT FOR: first_pass
Thu Feb 2 15:19:38 PST 2006 PAGE 1

1703009 1011 190 26 3a. Bottle # returned
jnl dick          060102 23:37 clean 1 = .D
dde 2nd           060103 10:42 clean 4 = .N

2106009 1000 395 52 E3. Total number SGS slides made
jnl harry         060103 09:57 clean 1 =
dde 2nd           060103 10:46 clean 4 = 02

1513919 2000 90 16 7. Years in school
jnl harry         060104 10:36 clean 1 = 16
dde 2nd           060110 12:13 clean 4 = 12

1503600 1070 200 8 Visit Date
jnl harry         060104 11:52 clean 1 = 04/01/05
dde 2nd           060122 11:37 clean 4 = 04/01/06
```



## DF\_compare

```
#!/usr/local/bin/perl

# Some housekeeping stuff, first argument in reports tool is study
# number and we need to supply a date range (-t) and 2nd pass
# validation level (-s). Also need to figure out the study directory.

$STUDY = shift(@ARGV);
use Getopt::Std;
getopt('ts');
$[ = 1;
die "Must supply -t option <yymmdd-yymmdd> date range" if (!$opt_t);
$DATES = $opt_t;
die "Must supply -s option for 2nd pass validation level" if (!$opt_s);
$VAL = $opt_s;
open (INFO, "DFgetparam.rpc -s $STUDY STUDY_DIR |")
  || die "DFgetparam.rpc -s $STUDY STUDY_DIR *FAILED*";
$STUDY_DIR = <INFO>;
```



## DF\_compare

---

```
# first call DFdde.set with "-new" option to grab 1st entry records
# which were entered during "-t" date range. "-remake" option
# allows overwrite of set "first_pass"

system("DFdde.set $STUDY -new -t $DATES -remake -set first_pass");

# Next call DFdde.set with "-v $VAL" option to grab 2nd pass records
# at the assigned 2nd pass level during the same "-t" date range

system("DFdde.set $STUDY -t $DATES -v $VAL -remake -set
second_pass");

# Need to delete the first_pass.dde file, otherwise DDE doesn't
# think second pass of data entry has even started

system("/bin/rm -f $STUDY_DIR/dde/sets/first_pass.dde");
```



## DF\_compare

---

```
# read in and keep track of all of the lines of the quasi-journal file
# representing the 1st pass of entry

open(FIRST_PASS, "$STUDY_DIR/dde/sets/first_pass.jnl") || die;
while(<FIRST_PASS>) {
    @data = split(/\|/, $_);
    $first_rasters{$data[7]} = $_;
}

# read in and keep track of all of the lines of the quasi-journal file
# representing the 2nd pass entry. While we're at it, replace the 2nd pass
# user name with just "2nd"

open(SECOND_PASS, "$STUDY_DIR/dde/sets/second_pass.jnl") || die;
while(<SECOND_PASS>) {
    @data = split(/\|/, $_); $data[3] = "2nd";
    $line = join("|", @data);
    $second_rasters{$data[7]} = $line;
}
```



## DF\_compare

---

```
# Almost done, but we need to better "line up" 1st pass and 2nd pass  
# entry, or DFdde.cmp later doesn't like it too much. Rewrite the  
# 1st pass data file with only records that have been 2nd passed
```

```
open(FIRST_DAT, "> $STUDY_DIR/dde/sets/first_pass.dat") || die;  
foreach $raster (keys(%first_rasters)) {  
    print FIRST_DAT $second_rasters{$raster};  
}  
close(FIRST_DAT);
```

```
# Now just run DFdde.cmp to compare entry, ignoring string fields,  
# and run DFdde.ss4 to summarize in a (sort of) nice table
```

```
system("DFdde.cmp $STUDY -text -set first_pass");  
system("DFdde.ss4 $STUDY -set first_pass");
```



## DataFax Remote Access in < 5 Minutes


---

- Well, 10 minutes if you type slowly...
- DF/Net uses a Citrix product called "GoToMyPC" ([www.gotomypc.com](http://www.gotomypc.com))
- For under \$20/month, allows remote desktop access to your PC at work
- Projects your desktop to anywhere



## As Easy as 1-2-3

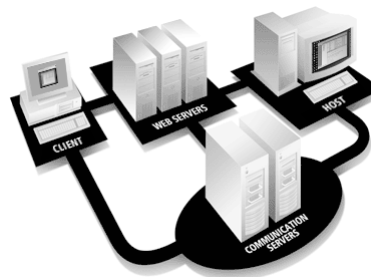


- Stumbled on it while trying to buy a remote solution from the difficult to navigate Citrix web site
- Followed a link to [www.gotomypc.com](http://www.gotomypc.com),  and as easy as 1-2-3, had it working!



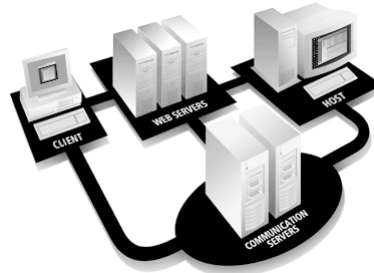
## How it Works

- Host (PC on your desk at work) is always running a small applet
- Host regularly sends outgoing “pings” to remote GoToMyPC web servers waiting for connection requests
- GoToMyPC uses firewall friendly ports 80, 443, and/or 8200
- Client (e.g., your laptop) authenticates to [www.gotomypc.com](http://www.gotomypc.com). Your remote connection is displayed using a Java-based or universal viewer



## How it Works

- Citrix web servers act as connection brokers to match host and client computers
- Client authenticates to host computer (SecurID optional)
- Traffic directed as end-to-end AES 128-bit encrypted data stream through the communication servers
- All the heavy lifting lives somewhere else, not in your office



## Advantages for Us

- The complicated stuff happens somewhere else - no connectivity hardware to buy, minimal software to install, upgrade and manage
- Access to your personal desktop as if you were sitting at work
- Low total cost of ownership

*But...*

- GoToMyPC implies 1 person, 1 PC



## Validation Tools Sizer

---

- We often access DataFax via X-windows on a variety of monitors:
  - 1920x1200 on the desktop
  - 1024x768 on the laptop
  - 1280x1024 on the conference room projector
- Wouldn't it be nice for the DataFax tools to automatically size correctly for each?



## Validation Tools Sizer

---

- A simple addition to your home ".cshrc" file will run xwininfo which lists various screen attributes. For example:
  - Absolute upper-left X: 0*
  - Width: 1920*
  - Height: 1200*
  - ... etc ...*
- From this, we can determine the width and height, and use "xrdb" to apply appropriate DataFax settings



## ~/.cshrc file additions

```
# Setting for DataFax tool sizing based on screen resolution

# xwininfo and xrdb reside in /usr/openwin/bin
set path=( $path /usr/openwin/bin )

# If you don't have a display, don't even bother
if ( ${?DISPLAY} ) then
    # get width and height from xwininfo
    set WIDTH=`xwininfo -root | grep Width | sed 's/ Width: //g'`
    set HEIGHT=`xwininfo -root | grep Height | sed 's/ Height: //g'`
    xrdb -merge ~/.Xdefaults-"$WIDTH"x"$HEIGHT"
endif
```



## .Xdefaults file contents

**==> .Xdefaults-1024x768 <==      ==> .Xdefaults-1400x1050 <==**

DFvalidate*.validateWindow.width: 895	DFvalidate*.validateWindow.width: 895
DFvalidate*.crfScrolledWindow.height: 330	DFvalidate*.crfScrolledWindow.height: 468
DFvalidate*.dataScrolledWindow.height: 330	DFvalidate*.dataScrolledWindow.height: 468
DFvalidate*.viewHorizButton.set: False	DFvalidate*.viewHorizButton.set: False
DFsetup*.setupWindowShell.width: 669	DFsetup*.setupWindowShell.height: 1015
	DFsetup*.setupWindowShell.width: 941

**==> .Xdefaults-1280x1024 <==      ==> .Xdefaults-1920x1200 <==**

DFvalidate*.validateWindow.width: 895	DFvalidate*.validateWindow.width: 1647
DFvalidate*.crfScrolledWindow.height: 455	DFvalidate*.crfScrolledWindow.height: 1107
DFvalidate*.dataScrolledWindow.height: 455	DFvalidate*.dataScrolledWindow.height: 1107
DFvalidate*.viewHorizButton.set: False	DFvalidate*.viewHorizButton.set: True
DFsetup*.setupWindowShell.height: 989	DFsetup*.setupWindowShell.height: 1165
DFsetup*.setupWindowShell.width: 941	DFsetup*.setupWindowShell.width: 941



## How Many Modems Do you Really Need?

---

- Very simple perl script to parse `/var/spool/fax/etc/xferfaxlog` for the answer
- Summarizes time used per modem for incoming/outgoing faxes
- Little or zero time used on the outer edges might point to low usage



## Modem Log Tools

---

```
term_a000: Total time = 267:39:49 (hh:mm:ss)
term_a001: Total time = 1121:24:38 (hh:mm:ss)
term_a002: Total time = 309:29:28 (hh:mm:ss)
term_a003: Total time = 39:53:37 (hh:mm:ss)
term_a004: Total time = 05:29:57 (hh:mm:ss)
term_a005: Total time = 05:29:29 (hh:mm:ss)
term_a006: Total time = 03:55:21 (hh:mm:ss)
term_a007: Total time = 03:28:45 (hh:mm:ss)
```





## I'd Rather Use a UNIX Editor (like vi)

---

- Simple vi macro to save, compile, and display errors. Add to your “.cshrc” file:  
setenv EXINIT 'map T :w!^M:\!DFcompiler -s 20 %^M'
- Edit the DFedits file in “vi”, and use Shift-T to emulate the “Check Syntax” editor function
- “T” writes the file, runs the “DFcompiler -s 20” (where 20 is a DF study number) and “%” is automagically replaced by the filename itself
- ^M's are entered with control-v control-m



## Questions?

---

Darryl Pahl

DF/Net Research, Inc.  
140 Lakeside Avenue, Suite 310  
Seattle, WA 98122

darryl@dfnetresearch.com

