Signature Tracking Program

# Introduction

The purpose of the signature tracking program is to evaluate which signatures have been invalidated by changes to data covered by the signature. The program works by reading a configuration file to learn where the study’s signature fields are and what plates and data fields each of them covers. The audit trail is then processed to determine when records were signed and to track changes to data fields covered by that signature after the signature was executed. When the program has finished reading all of the audit trail, it then knows which records were never signed, which have valid signatures and which have invalidated signatures and can output this list to an Excel spreadsheet and/or DataFax DRF file.

# Configuration File Format

In order to track electronic signatures, the program needs to know where they are. This is accomplished via a study specific configuration file. A configuration file can contain entries for multiple signatures and the plates and data fields each of them covers. The configuration file is a text file with the following format:

signature “***SIGNATURE\_NAME***” plate ***PLATE\_NUMBER*** visit ***VISIT\_RANGE*** fields ***SIGNATURE\_FIELDS*** {  
 plate ***COVERED\_PLATE*** ignore fields ***IGNORED\_FIELDS***;  
 plate ***COVERED\_PLATE*** ignore fields ***IGNORED\_FIELDS***;  
 …  
}

where:

***SIGNATURE\_NAME*** = a descriptive name for the signature (e.g. Randomization)  
***PLATE\_NUMBER*** = plate containing signature  
***VISIT\_RANGE*** = range of visits that this plate can occur at  
***SIGNATURE\_FIELDS*** = field numbers of the signature fields  
***COVERED\_PLATE*** = a plate number covered by this signature  
***IGNORED\_FIELDS*** = fields on the plate that, if changed, do not require a re-sign (e.g. coding fields, HELP)

A ***COVERED\_PLATE*** statement is required for each plate covered by the signature. The ignore fields clause can be omitted if all data fields on the plate are important for the integrity of the signature.

A study may have multiple signatures as shown in the next example:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Plate | Visit | Sig. Fields | Covered Plate | Coding Fields |
| Randomization | 4 | 1 | 19-23 | 1 | 32 |
| 2 | 26 |
| 3 | 31 |
| 4 | 24 |
| Hospitalization | 458 | ANY | 41-45 | 458 | 46,47-49 |

The corresponding configuration file would look like this:

signature "Randomization" plate 4 visit 1 fields 19-23 {  
 plate 1 ignore fields 32;  
 plate 2 ignore fields 26;  
 plate 3 ignore fields 31;  
 plate 4 ignore fields 24;  
}  
signature "Hospitalization" plate 458 visit \* fields 41-45 {  
 plate 458 ignore fields 46,47-49;  
}

Note the use of \* in the visit parameter to indicate any visit.

Care must be taken to keep the configuration file up to date with changes in the study setup. If the signature fields or the ignored fields change field numbers, then the configuration file must be updated accordingly.

# Running the Signature Tracking Program

The signature tracking program can be run once a suitable configuration exists. The signature tracking program reads and processes the output of the DataFax DFaudittrace program to determine what fields have changed and thereby signature validity. The signature tracking program has options to generate an Excel file containing the statuses of all signature plates and a DRF file with the keys of records with invalidated signatures.

A command to process the audit trail and generate an Excel and DRF file would be:

$DATAFAX\_DIR/bin/DFaudittrace -s $STUDY \  
 -N -d 19900101-today | sigtrack \  
 --config $STUDY\_DIR/work/sigtrack.config \  
 --xls $STUDY\_DIR/work/esig.xlsx \  
 --drf $STUDY\_DIR/drf/esig.drf

The amount of output produced by DFaudittrace for large studies can be massive so additional filters may be added to reduce the output to only records involved in the signature and thereby speed up the process. In our example configuration file, only plates 1-4 and 458 are involved in determining signature state and we may further want to exclude test patient IDs from 998000 to 999999. The command line would then be changed to:

$DATAFAX\_DIR/bin/DFaudittrace -s $STUDY \  
 **-I 0-997999,1000000-2147999 –P 1-4,458** \  
 -N -d 19900101-today | sigtrack \  
 --config $STUDY\_DIR/work/sigtrack.config \  
 --xls $STUDY\_DIR/work/esig.xlsx \  
 --drf $STUDY\_DIR/drf/esig.drf

The DFaudittrace options are:

-s study number  
-I patients to include  
-P plates to include  
-N include new record detail (this option is required for the tracking program)  
-d date ranges to report on

The sigtrack options are:

--config configuration file name  
--xls generate an Excel spreadsheet  
--drf generate a DRF file of records with invalidated keys  
--allow-signer-changes allow a signer to change the record without requiring a re-sign  
--resign-when-final put entries in DRF file only if all covered records are at final state  
--sdv use labels more suitable for SDV flag dates in Excel files  
--studydir DataFax study directory. Used to find DFcenters and DFcountries  
--db generate a signing database file for later PDF generation  
--version print sigtrack version number

The tracking process can be run interactively as a DataFax report or via cron if it is to be done on a periodic basis.

# Excel Output File

The Excel output file has several columns with auto-filters on them:

1. Region – The region this patient is in (hidden unless --studydir is specified)
2. Country – The country this patient is in (hidden unless --studydir is specified)
3. Center – The center this patient is in (hidden unless --studydir is specified)
4. Patient ID – The patient ID
5. Visit – The visit number for this signature
6. Sig. Plate – The plate number for this signature
7. Sig. Desc – The plate description for this signature
8. Status – The status of the signature
9. Signer – The username of the person who signed the record
10. Signed – The date and time the signature was executed (server time)
11. Plate – The plate number of the plate covered by that signature
12. Field – The field number on the above plate that was changed
13. Description – The field description for the above field
14. Signed Value – The value the field had at signature time
15. Current Value – The value the field now has
16. Last Changer – The username of the last user to change that field
17. Last Changed – The date and time the last change happen at (server time)
18. Comment – A comment to describe the change (e.g. changed by signer)

There will be a row for each field changed that is covered by a signature.

The status field can have several values:

|  |  |  |
| --- | --- | --- |
| Status | Color | Meaning |
| NEVER SIGNED | Yellow | The record has never been signed |
| UNSIGNED ERROR RECORD | Purple | The record is at level 7 with status pending but was never signed |
| UNSIGNED LOST RECORD | White | The record is marked lost but has never been signed |
| UNSIGNED DELETED RECORD | Dark Red | The record was deleted from the database but was never signed |
| SIGNATURE REMOVED | Red | The record was signed, but one or more signature fields were blanked out |
| SIG. REMOVED, ERROR RECORD | Purple | The record was signed, one or more signature fields were blanked out and then the record was marked as Error (level 7 pending) |
| SIG. REMOVED, LOST RECORD | White | The record was signed, one or more signature fields where blanked out and then the record was marked as lost |
| SIG. REMOVED, DELETED RECORD | Dark Red | The record was signed, one or more signature fields were blanked out and then the record was deleted |
| SIGNATURE OK | Green | The record was signed and no changes to affect the signature have been made |
| DATA CHANGE BY SIGNER | Green | The record was signed and was subsequently changed by the person who signed it (only valid with –allow-signer-changes option) |
| RE-SIGN REQD | Dark Red | The record was signed and was subsequently changed and requires a re-sign |
| RE-SIGN REQD WHEN FINAL | Orange | This record was signed and was subsequently changed and requires a re-sign when all covered plates are at final status. Applies only if ‑‑resign-when-final option is specified on command line. |
| SIGNED IN ERROR | Purple | The record was signed and then was marked as an error record (level 7 pending) |
| SIGNED, MARKED LOST | White | The record was signed and then was marked as a lost record |
| DELETED SIGNED RECORD | Dark Red | The record was signed and was subsequently deleted |

In –sdv mode, the following labels are used:

|  |  |  |
| --- | --- | --- |
| Status | Color | Meaning |
| NEVER VERIFIED | Yellow | The record has never been source verified |
| NEVER VERIFIED (ERROR REC) | Purple | The record is at level 7 with status pending but was never source verified |
| NEVER VERIFIED (LOST REC) | White | The record is marked as lost but was never source verified |
| NEVER VERIFIED (DELETED REC) | Dark Red | The record was deleted from the database but was never source verified |
| RE-VERIFICATION REQD | Red | The record was verified, but needs re-verification because of data changes |
| RE-VERIFICATION REQD (ERROR REC) | Purple | The record was verified, data changes were made, and then the record was marked as Error (level 7 pending) |
| RE\_VERIFICATION REQD (LOST REC) | White | The record was verified, data changes were made, and then the record was marked as lost |
| RE-VERIFICATION REQD (DELETED REC) | Dark Red | The record was verified, data changes were made, and then the record was deleted |
| SDV OK | Green | The record was source verified and no changes to affect the data have been made |
| RE-VERIFICATION REQD | Green | The record was verified and was subsequently changed by the person who verified it (only valid with –allow-signer-changes option) |
| RE-VERIFICATION REQD | Dark Red | The record was verified and was subsequently changed and requires a re-verification |
| RE-VERIFICATION REQD WHEN FINAL | Orange | This record was verified and was subsequently changed and requires a re-verification when all covered plates are at final status. Applies only if ‑‑resign-when-final option is specified on command line. |
| SDV OK (ERROR REC) | Purple | The record was verified and then was marked as an error record (level 7 pending) |
| SDV OK (LOST REC) | White | The record was verified and then was marked as lost |
| SDV OK (DELETED REC) | Dark Red | The record was verified and was subsequently deleted |

# Regions and Countries

The Excel report has the ability to write out the region, country and site information for each record. To do this, the program must be run with the --studydir option so that it can locate the DFcenters and DFcountries file in the study lib directory. The DFcenters file is part of DataFax, while the DFcountries file is PHRI specific.

The format of the DFcountries file is:

Country|Region|Center Ranges

For example, if centers 10-19 are in Canada, 20-29 are in USA and 30-39 are in Germany, then the DFcountries file might look as follows:

Canada|North America|10-19  
USA|North America|20-29  
Germany|Europe|30-39

E-Signature PDF Generation

# Introduction

The E-Signature PDF generation process generates a PDF file for each patient showing the values of critical fields at the time of signing. Each signature shows the server timestamp when it was executed, along with the signature description, user who committed the signature to the database, visit and plate details. This information is followed by any new or changed data values and the values of the signature fields on the CRF.

The generation process is done in two parts. The first is determining when records where signed and what the data values were at signing and storing them in an intermediate database. The second phase reads that database and generates the PDF files themselves.

# Generating the Intermediate Database

The process of generating the intermediate database involves running the signature tracking report/program with the --db option to specify the output database name in addition to any other options normally used.

# Generating the PDF files

Once the intermediate database has been generated, the task of generating the output PDF files can begin. This is accomplished by running the signature.py program which generates a PDF file for each patient containing signature details and data values.

The signature.py program has many options to filter and customize its output.

|  |  |
| --- | --- |
| --db *database* | The name of the intermediate database to use. Default is ‘data.db’. |
| --studydir *path* | (Required) The path to the DataFax $STUDY\_DIR. |
| --ids *id-list* | The list of patient IDs to include. Default is to include all patients. |
| --plates *plate-list* | The list of plate numbers to include. Default is to include all plates. |
| --visits *visit-list* | The list of visit numbers to include. Default is to include all visits. |

For example, to create a PDF file for patient 99001 you would execute:

python27 signature.py --studydir /opt/val254 --ids 99001

An output file named 99001.pdf will be created in the current directory.