TCRN STANDARD OPERATING PROCEDURE

TITLE:	NUMBER: SOP-1 Version: 11
Validation of De-Identification Quality	
PREPARED BY:	**APPROVED BY:**
Monica Lopez Murphy, Roswell Park Cancer Institute	**TCRN Executive Committee**
DATE WRITTEN:	**ISSUE DATE:**
March 9, 2015	**March 10, 2015**

1.0 PURPOSE:

This policy describes the guidelines and process by which the TIES Cancer Research Network (TCRN) will conduct validation of the de-identification of the pathology reports loaded into the TIES system which have been de-identified using the De-IData software.

2.0 REVISION HISTORY:

Date	Rev. No.	Modification
6/2/14	1	Original Document authored by Monica Lopez Murphy
6/22/14	2	Minor Changes by Rebecca Crowley Jacobson
6/24/14	3	Modification to Procedure section by Monica Lopez Murphy
6/30/14	4	Minor Changes by Rebecca Crowley Jacobson
7/02/14	5	Minor changes and edits from Subcommittee Members
7/15/14	6	Modification to distinguish Load QA from Ongoing QA
10/21/14	7	Minor Changes by Liz Legowski
10/23/14	8	Minor Changes by Monica Lopez Murphy
3/5/15	9	Minor Changes by Liz Legowski to use wording in revised network agreement
3/5/15	10	Modifications by Julia Corrigan and Liz Legowski
3/9/15	11	Minor modifications by Liz Legowski

3.0 PERSONS AFFECTED:

Quality Assurance personnel at institutions participating in TCRN.

4.0 POLICY:

Each TCRN member institution will conduct a validation of the de-identification of reports loaded into TIES. Validation and QC of the de-identification process will occur as part of the system testing for initial validation of the system following the loading of reports. Each Institution will also perform the Quality Assurance testing at institution-defined time points with a **minimum frequency of once per year**.

5.0 DEFINITIONS:

TIES – TIES (Text Information Extraction System) is a computer-based system that establishes a repository of natural language processing (NLP) coded, de-identified pathology reports for the purpose of identifying cohorts and cases associated with formalin-fixed paraffin-embedded materials, frozen tissues, or other research resources.

TCRN – The TIES Cancer Research Network (TCRN) represents all member institutions that have signed the TCRN network agreement, with the intent of supporting collaboration (data, tissue, or data and tissue) across institutions that have deployed the TIES system.

DeID – The commercial software program used in the TIES system by TCRN members for the purpose of text de-identification of electronic health record free text data.

6.0 RESPONSIBILITIES:

De-identification validation requirements at each TCRN institution will be determined by that institution's site PI or the TCRN **Quality Assurance Manager**.

It is the responsibility of each TCRN Institution's Quality Assurance Manager to ensure that reports loaded into the TIES system are appropriately de-identified. An initial validation of the system will be performed once all of the reports are loaded. Subsequently, ongoing Quality Assurance checks will be performed at regular intervals. Each institution is expected to determine the frequency of ongoing QA checks of its own TIES node. However, **such ongoing QA checks should occur at least once per year**.

The results of the QA checks will be entered into the TCRN QA database and a report will be generated by each institution's QA manager and will be shared with the TCRN Executive Committee on an annual basis.

7.0 PROCEDURES:

Determine Validation Requirements for the Institution (site PI or QA Manager):

1. Engage the stake holders at the institution to determine institution-specific requirements for validation and documentation. It is expected that individual institutions may set institution-specific thresholds for assuring sufficient de-identification, intervals for QA, or other requirements.
2. Delineate requirements for initial de-identification Quality Assurance done after completing a bulk load of retrospective documents (also known as Load QA) as distinguished from periodic interval de-identification Quality Assurance (also known as Ongoing QA).
3. For both Load QA and Ongoing QA, determine any thresholds for de-identification (e.g. frequency of PHI elements, etc).
4. For Ongoing QA, determine the frequency of QA checks required by the institution.
5. For both Load QA and for Ongoing QA, determine any requirements for randomization in report selection.
6. Create a document outlining data contained in the reports loaded into TIES that may be viewed as PHI or could be used to identify a patient. In this document, delineate whether each data item is considered PHI and whether the system is expected to remove or tag it. This document will be used by the QA managers and their teams to validate the de-

identification process. See example in *Appendix A*. An explanation of all deID tags is provided in *Appendix B*.

Determine Number of Reports to be Validated (QA Manager):

A. For both Load QA and for Ongoing QA, enumerate the number of reports being loaded into TIES that require QA.
B. If the institution's IRB or Legal Counsel requires that a specific methodology be used (e.g. sample size calculation), define the steps for determining number accordingly.
C. For sample size determination, determine the number of reports that need to be manually checked for appropriate de-identification by entering the following criteria into the sample size calculator found at: https://www.mccallum-layton.co.uk/tools/statistic-calculators/sample-size-calculator/
    i. Margin of error that is acceptable
    ii. Population size = total number of reports in the system that require QA
    iii. Select a confidence level of 95% or above
    iv. Use the calculated value to set the institution's criteria for the number of reports that will require manual validation of de-identification

Validation of de-Identification in TIES for both Load QA and Ongoing QA (QA Manager):

1. The QA manager logs in to TIES as an Honest Broker and runs a query that will result in the return of all the reports which require validation (e.g. reports from a specific year). Check to ensure number of reports returned is at least the number that the institution is required to validate.
2. Re-run the query checking the "Randomize" checkbox on the right hand side of the query interface. Also indicate how many reports the institution would like returned – this number needs to be larger than the number of reports required to be validated (calculated above). Using the built-in randomize function will return a randomized subset of reports from the total set required for validation. Further information about TIES randomization is available in *Appendix C*.
3. Create a case set folder appropriately labeled as QA set with a description of the reports being checked (e.g.QA CY2002pathology reports).
4. Choose reports that will be checked by clicking and dragging them into a case set.
5. Create an Excel output of all cases to be checked and use that Excel document to record the following information:
    A. Report de-identification
        1. Pass/Fail/Over scrubbing
    B. Failure Detail –
        1. Data item
        2. Issue
    C. Notes – use this field to add any comments or notes regarding the findings

    An example of this document is shown in *Appendix D*.
6. Proceed with the de-identification QA by reviewing all chosen reports in the case set. Using the TIES PHI list of allowable data elements, review each report contained in the case set. Record a Pass if no PHI or PHI-like data remained in the pathology report, and Fail for those reports where a data item that should have been removed was not. For failed reports, please record the data item that was missed and what the reason or issue was, if it can be assessed.

7. Once all reports have been validated and results have been recorded, enumerate and generate a summary count of the results.
8. Enter the summary data into the TIES Quality Assurance database.
9. Review results with institutional stakeholders. Determine the level of risk that each failure generates and whether it is necessary to address the failure in the system.
10. The TCRN Executive Committee will regularly review results of all Load and Ongoing QA studies.

Managing Failures

1. Immediately quarantine any failed reports.
2. Each institution will work with the deID vendor to mitigate any failures believed to apply to multiple documents.
3. Documents that have been quarantined will be reprocessed to remove any PHI and checked to ensure that all PHI has been removed.
4. For documents that do not warrant major change to the deID software or configuration files, QA personnel will manually redact any PHI prior to reintroducing the document into circulation in the TIES system.

8.0 APPENDICES:

A. Data List of PHI in report template
B. De-IDATA table of deID tags
C. Randomization schema
D. QA result report template

REFERENCES:

None

Appendix A

Example Template for Data List of PHI

Data Item(s)	Description (If needed)	Section location	PHI Y/N	Necessary to Remove	Notes	De-ID for PI view
Address	Patient Address	Header	Y	Y	Not loaded into TIES	
Age		Clinical History	N	N		changes it to indicate decade
Age	Patient Age	Header	N	N		True Pt age
Call time	Call time from one MD to another MD		N	N		
Date	Dates references in clinical history	Clinical History	Y	Y		Off set the date
Date	Date the report was generated	Header	Y	Y		Off set the date
Date	Electronically Signed Date	Results	Y	Y		Off set the date
Dates of other tests		Results	Y	Y	such as a CBC	Off set the date

Data contained in this template is for illustration purposes only

Appendix B – De-ID Tags

	TAG	COMMENT
Names:	**NAME[AAA], **NAME[BBB CCC]	The name tage includes a set of letters indicating that the name has been replaced with the tag "**NAME[AAA]" and another name with **NAME[BBB,CCC]". If the first name is repeated later in the document, it will be replaced again with "**NAME[AAA]." Caveat: If the same name repeats, even when belonging to a different person, the same letters will be used to denote that name. For example, Robert Johnson and Robert Williams may be replaced with "**NAME[AAA BBB]" and **NAME[AAA CCC]," respectively.
Geographical:	**PLACE	Names of US cities and towns are replaced with the tag **PLACE.
	**INSTITUTION	Names of businesses and/or building (such as Children's Hospital") are replaced with this tag.
	**STREET-ADDRESS	Street numbers and names are replaced with this tag.
	**ZIP-CODE	The full zip code is replaced no matter how large the population of the area happens to be.
Dates:	**DATE	Dates are offset by a specified amount per patient. The date offset differs between patients based on the patients' ID numbers. Since ID numbers must be removed from de-identified reports, it is not possible for the end user to determine the offset based on this information. Also, since ID numbers do not (usually) change for a patient during her treatment, the timeline of events within this treatment should remain consistent.
Age:	**AGE	De-ID designates "age" as the decade in which the actual age occurs. A 35-year-old will be given the tag **AGE[in 30s]. There are three exceptions to this rule. Two of these are for children where the ranges "**AGE[birth-12]" and "**AGE[in teens]" are used; and one is for the elderly where "**AGE[90+]" is used.
Phone:	**PHONE	Telephone numbers are replaced with the tag **PHONE.
Fax:	**PHONE	Fax numbers are not distinguished from telephone numbers, and use the same tag.
Email:	**EMAIL	De-ID finds email addresses in any of the following domains: .com,. net, .gov, .edu and .org.
SSN:	**ID-NUM	
Medical Rec #:	**ID-NUM	
Insurance #:	**ID-NUM	
Other IDs:	**ID-NUM	
License #:	Not applicable	Due to lack of training data, De-ID does not currently handle license numbers.
Vehicle:	Not applicable	Due to lack of training data, De-ID does not currently handle vehicle identifiers.
Device:	**DEVICE-ID	De-ID removes serial and model numbers, but not model names.
URL:	**WEB-LOC	De-ID finds web addresses in any of the following domains: .com,. net, .gov, .edu and .org.
IP#:	**WEB-LOC	De-ID finds IP#s of four numeric components from 0 to 255.
~~Biometric Identifiers:~~		Biometric Biometric identifiers are not a part of free-text reports and are not handled by De-ID.
~~Images:~~		Images Images are not a part of free-text reports and are not handled by De-ID.
Other	**PATH-NUMBER (n)	Pathology specimen numbers are replaced with this tag. If the same number is referenced in the document, we sequentially number the tag so that the reader can track the same specimen being referenced in the document.

Appendix C
TIES Randomized Search Implementation

Overview

When TIES performs a search with the randomized payload feature checked it returns a random subsequence from the overall score sorted search hit batch. So if the query matches
u v w x y

in score sorted order a non randomized request for two documents would return
(u v)

Randomizing might return any of
(u v), (u, w) (u, x) (u, y), ... (x y)

Subsequence Selection

The algorithm used to winnow this random subset from a sorted set of numbers is from Jon Bently.

Title: Programming Pearls by Jon Bentley
Paperback: 256 pages
Publisher: Dorling Kindersley Pvt Ltd; 2nd edition edition (December 1, 2006)
Language: English
ISBN-10: 8177588583
ISBN-13: 978-8177588583
bitsortgen.c

```
/* bitsortgen.c -- gen $1 distinct integers from U[0,$2) */

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define MAXN 2000000
int x[MAXN];

int randint(int a, int b)
{       return a + (RAND_MAX * rand() + rand()) % (b + 1 - a);
}

int main(int argc, char *argv[])
{       int i, k, n, t, p;
        srand((unsigned) time(NULL));
        k = atoi(argv[1]);
        n = atoi(argv[2]);
        for (i = 0; i < n; i++)
                x[i] = i;
        for (i = 0; i < k; i++) {
                p = randint(i, n-1);
                t = x[p]; x[p] = x[i]; x[i] = t;
                printf("%d\n", x[i]);
        }
        return 0;
}
```

Pseudo Random Number Generation

At the heart of TIES implementation of the Bentley algorithm is the standard pseudo random number generator provided by Java. Pseudo random number generators all work the same way. Given a seed number they will generate a stream of subsequent numbers that "mimic" a truly random sequence.. This sequence is repeatable by rerunning the generator with the same seed value. TIES employs the standard computational practice of seeding the Java pseudo random number generator with the current timestamp. So the time of day it is when a user issues the query will initiate the random stream.

Documentation on the Java Random number generator is as follows.

Sets the seed of this random number generator using a single long seed. The general contract of setSeed is that it alters the state of this random number generator object so as to be in exactly the same state as if it had just been created with the argument seed as a seed.

The method setSeed is implemented by class Random by atomically updating the seed to
(seed ^ 0x5DEECE66DL) & ((1L << 48) - 1)
and clearing the haveNextNextGaussian flag used by nextGaussian.
The implementation of setSeed by class Random happens to use only 48 bits of the given seed. In general, however, an overriding method may use all 64 bits of the long argument as a seed value.

Parameters:
seed the initial seed

Appendix D

Failure Detail

	Patient MRN	Accession No.	Deidentified ID	Assessment	Data Item	Failure Reason	Notes
1	123456	S1999999		Value of Pass/ Fail / overscrubbing			

Data contained in this template is for illustration purposes only