IRI FieldShield

PII / PHI Classification & Masking

Product Overview

Technical Summary, Samples, and Specifications

IRI

Total Data Management

Table of Contents

Introduction

Consumers, patients, students and others expect their personally identifiable information (PII) and other sensitive data to remain protected from exposure, misappropriation and misuse. The companies and government agencies who collect, store, and/or process PII and other sensitive data are obliged to not only keep faith with these individuals, but assume the risks of financial liability or damage to their reputations for privacy law violations and data breaches.

The need to protect this data at risk -- at rest or in flight -- remains especially acute in traditional, transactional sources of data. Typically these are relational databases and structured files stored locally, or increasingly, in the cloud. They may also be used in testing and development.

One of the best ways to safeguard this data is by masking it through functions applied at the column level. This approach obfuscates data in secure and compliant ways that can nullify the effects of a data breach, differentiate access to original values, protect downstream applications from aforementioned risks, support auditing and still leave data usable for testing and analysis.

IRI FieldShield® is a robust data masking product for PII in databases and files. It was designed to help you adhere to both business rules and data privacy laws in a consistent, deterministic manner, either as a standalone solution or embedded in ETL, DevOps, or bespoke applications.

FieldShield is also a content-aware data loss prevention tool because it helps you find and apply the right 'shields' to only those fields that need masking. These field-level masks are also safer because even a breach of one field or encryption key still leaves remaining columns safe with their different protections. Contrast that to single password access to a full data source or silo.

Operations

Most FieldShield users start with integrated data classification and discovery operations. They then create and run -- automatically or manually in an Eclipse IDE called IRI Workbench -- data masking tasks or batch jobs (for multiple sources at once), with or without using data classes or masking rules, and without relying on formally defined schema constraints.

Masking jobs are ultimately defined in FieldShield Control Language (.fcl) scripts, which use the same (SortCL) program syntax as IRI CoSort® for transformation, cleansing, and reporting, IRI NextForm® for data and database migration, and IRI RowGen® for synthesizing test data. FieldShield and all of the above SortCL spinoffs are included along with visual ETL and other big data management features in the IRI Voracity® platform.

FieldShield library functions documented in an SDK can also be used for dynamic data masking or unmasking, and to embed data protections into distributed applications or SQL calls.

For PII in semi-structured and unstructured document and file formats, including free form text, MF Office documents,.pdf's and images, and multiple NoSQL DBs, IRI DarkShield® can be used. Both FieldShield and DarkShield can also find and mask PII stored in Excel spreadsheets, but IRI CellShield® is purpose-built for operating on data directly within Excel. The data masking functions are the same between the tools, so the ciphertext should remain consistent across the enterprise if you use more than one IRI masking tool..

Architecture

FieldShield works on – and can address DB and file sources in – LAN or cloud systems running Linux, Unix or Windows.

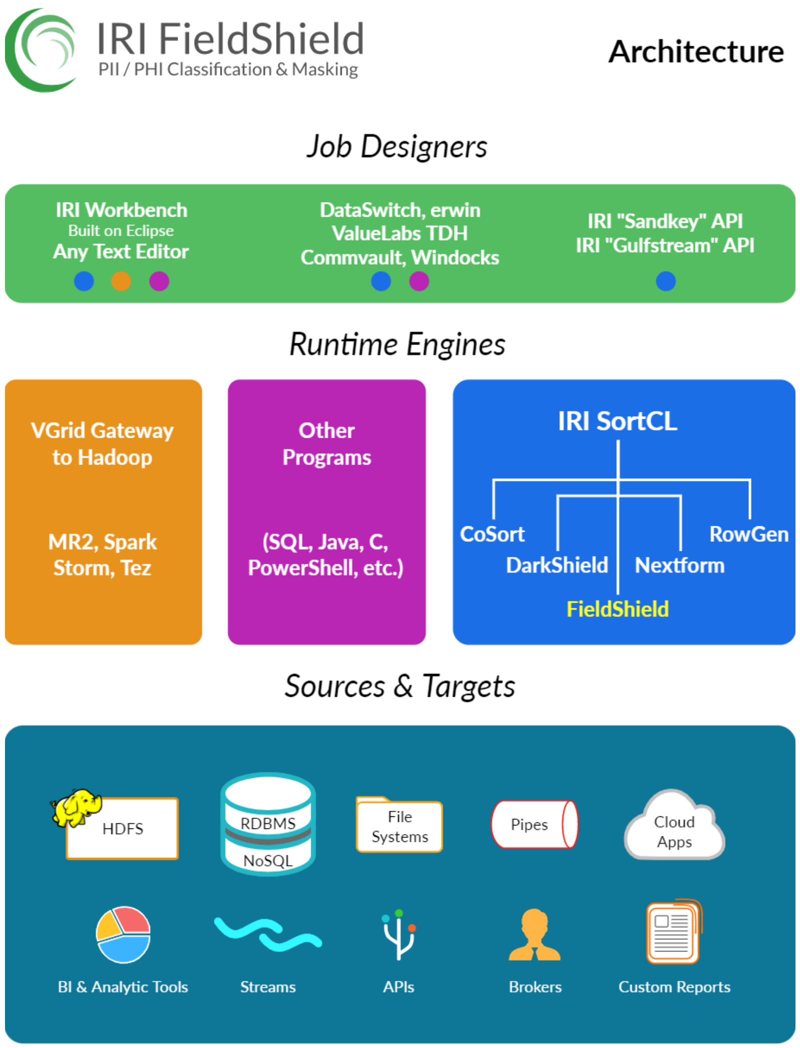
FieldShield uses IRI Workbench for client-side data discovery, and design of data-masking jobs, which serialize as portable text scripts that run from Workbench or any command line interface.

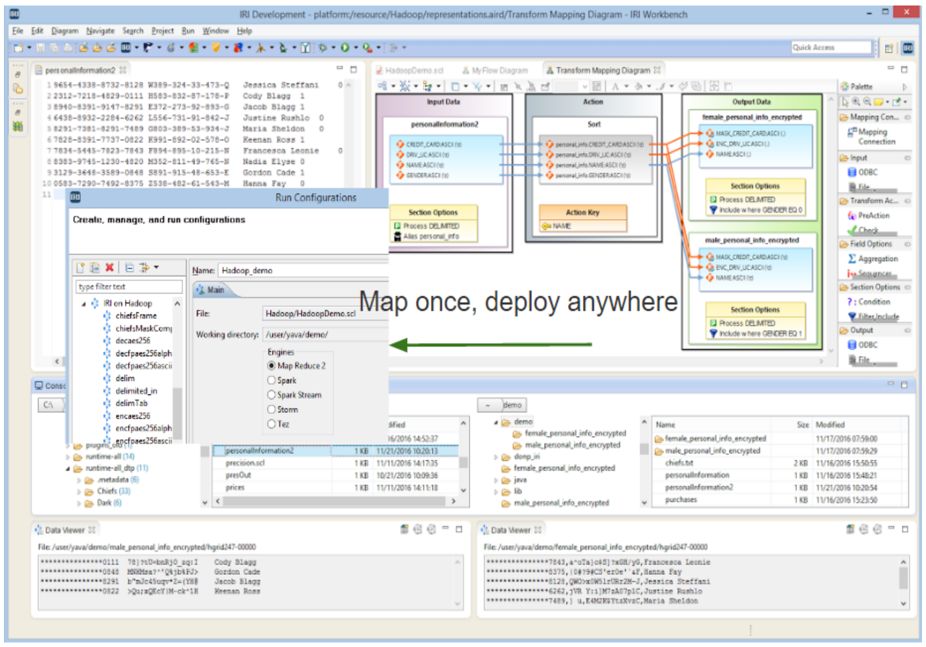
Many of the same scripts also run interchangeably in Hadoop MR2, Spark, Spark Stream, Storm or Tez for users of the big data edition of IRI Voracity.

FieldShield metadata is also fully supported in Workbench data models (EMF and XMI in Eclipse), and by Erwin Mapping Manager. Both offer graphical job creation, modification, and management.

Separate Workbench wizards exist to discover PII, create metadata, and risk score the re-identifiability of data sets.

FieldShield masking functions, such as encryption, hashing, and redaction, can also be called into application programs written in C/C++, Java, and .NET through libraries in a separately available SDK.




IRI Development - platform:/resource/Hadoop/representations.aird/Transform Mapping Diagram - IRI Workbench
File Edit Diagram Navigate Search Project Run Window Help
Transform Mapping Diagram
Input Data
Action
Output Data
personalinformation2
Sort
female_personal_info_encrypted
male_personal_info_encrypted
Section Options
Action Key
Run Configurations
Create, manage, and run configurations
Name: Hadoop_demo
Main
File: Hadoop/HadoopDemo.scl
Working directory: /user/yava/demo/
Engines
Map Reduce 2
Spark
Spark Stream
Storm
Tez
Map once, deploy anywhere
Data Viewer
Palette

PII Discovery

FieldShield's included data profiling and classification tools, pattern searches, forensic metadata discovery, and metadata management systems support data masking and governance by exposing the locations and attributes of PII before and after protection.

These tools are provided in IRI Workbench, the graphical IDE for FieldShield data discovery and masking jobs, built on Eclipse™. Included wizards support the discovery and definition of PII across disparate data sources, including DB tables, flat files, and dark (unstructured) data sources. Multiple reports from these search logs are produced, as are interactive dashboards.

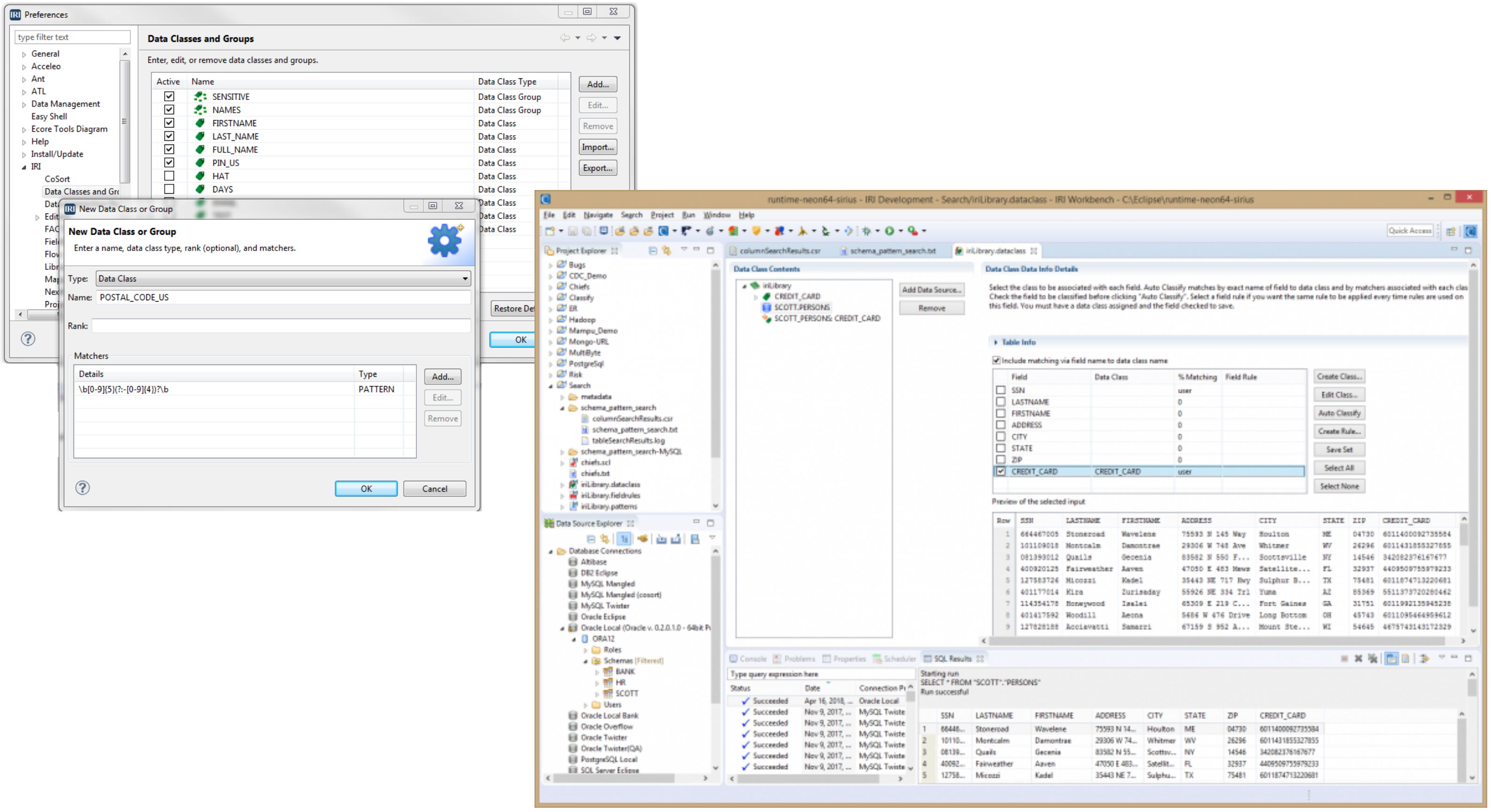
These discovery processes can also support data cleansing, integration (ETL), subsetting, reformatting, test data generation, data wrangling, and reporting. Unique among data masking tools is this easy and guarded secret of performing such functions in conjunction with, or directly within, FieldShield masking and other metadata-compatible IRI data management operations.

Data Classification & Masking

Define and manage enterprise-wide data class libraries, and use them to search and mask data across multiple sources at once. For more information, read this end-to-end how-to article::

https://www.iri.com/blog/data-protection/classify-mask-pii-in-databases-with-fieldshield/

or watch the video: www.youtube.com/watch?v=7nxp3TjsCqU&ab_channel=IRITheCoSortCo



Data Classes and Groups

Enter, edit, or remove data classes and groups.

Active	Name	Data Class Type
☑	SENSITIVE	Data Class Group
☑	NAMES	Data Class Group
☑	FIRSTNAME	Data Class
☑	LAST_NAME	Data Class
☑	FULL_NAME	Data Class
☑	PIN_US	Data Class
☐	HAT	Data Class
☐	DAYS	Data Class

New Data Class or Group

Enter a name, data class type, rank (optional), and matchers.

Type: Data Class
Name: POSTAL_CODE_US
Rank:

Matchers

Details	Type
\b[0-9]{5}(?:-[0-9]{4})?\b	PATTERN

Select the class to be associated with each field. Auto Classify matches by exact name of field to data class and by matchers associated with each clas Check the field to be classified before clicking "Auto Classify". Select a field rule if you want the same rule to be applied every time rules are used on this field. You must have a data class assigned and the field checked to save.

☑ Include matching via field name to data class name

	Field	Data Class	% Matching	Field Rule
☐	SSN		user	
☐	LASTNAME		0	
☐	FIRSTNAME		0	
☐	ADDRESS		0	
☐	CITY		0	
☐	STATE		0	
☐	ZIP		0	
☑	CREDIT_CARD	CREDIT_CARD	user	

Preview of the selected input

Row	SSN	LASTNAME	FIRSTNAME	ADDRESS	CITY	STATE	ZIP	CREDIT_CARD
1	664467005	Stoneroad	Wavelene	75593 N 145 Way	Houlton	ME	04730	6011400092735584
2	101109018	Montcalm	Damontrae	29306 W 748 Ave	Whitmer	WV	26296	6011431855327855
3	081393012	Quails	Gecenia	83582 N 550 F...	Scottsville	NY	14546	342082376167677
4	400920125	Fairweather	Aaven	47050 E 483 Mews	Satellite...	FL	32937	4409509755979233
5	127583726	Micozzi	Kadel	35443 NE 717 Hwy	Sulphur B...	TX	75481	6011874713220681
6	401177014	Kira	Zuriseday	55926 NE 334 Trl	Yuma	AZ	85369	5511373720280462
7	114354178	Honeywood	Izalei	65309 E 219 C...	Fort Gaines	GA	31751	6011992135945238
8	401417592	Woodill	Aeona	5686 W 476 Drive	Long Bottom	OH	45743	6011095464959612
9	127828188	Acciavatti	Samarri	67159 S 952 A...	Mount Ste...	WI	54645	4675743143172329

Type query expression here

Status	Date	Connection Pr
✓ Succeeded	Apr 16, 2018, ...	Oracle Local
✓ Succeeded	Nov 9, 2017, ...	MySQL Twiste
✓ Succeeded	Nov 9, 2017, ...	MySQL Twiste
✓ Succeeded	Nov 9, 2017, ...	MySQL Twiste
✓ Succeeded	Nov 9, 2017, ...	MySQL Twiste
✓ Succeeded	Nov 9, 2017, ...	MySQL Twiste
✓ Succeeded	Nov 9, 2017, ...	MySQL Twiste

```
Starting run
SELECT * FROM "SCOTT"."PERSONS"
Run successful
```

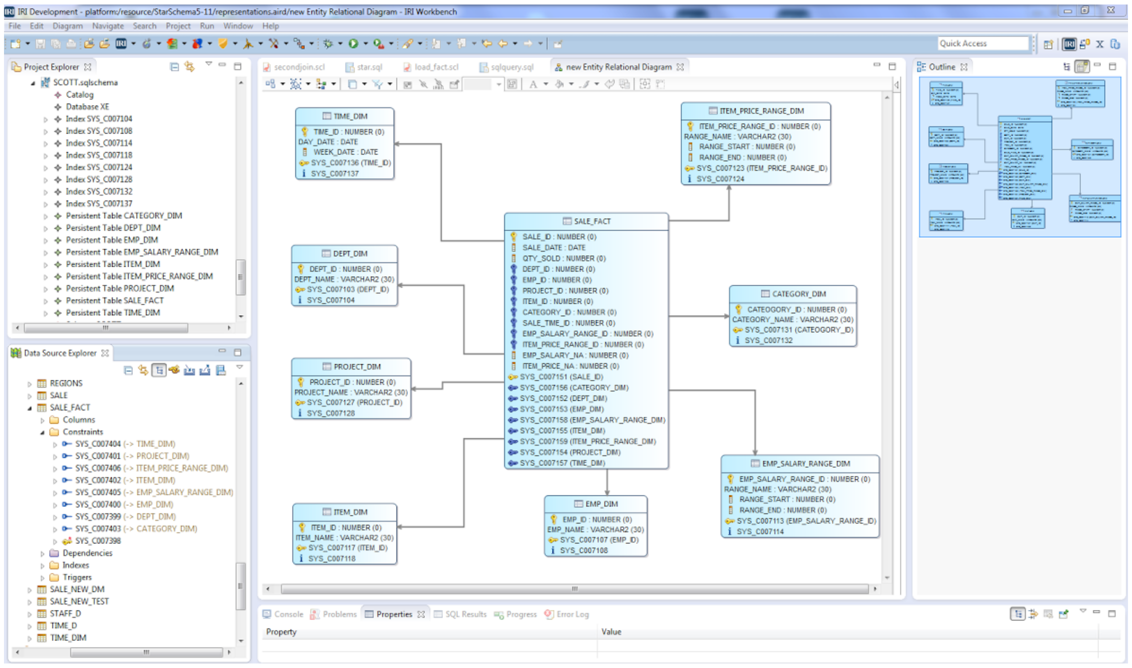
	SSN	LASTNAME	FIRSTNAME	ADDRESS	CITY	STATE	ZIP	CREDIT_CARD
1	66446...	Stoneroad	Wavelene	75593 N 14...	Houlton	ME	04730	6011400092735584
2	10110...	Montcalm	Damontrae	29306 W 74...	Whitmer	WV	26296	6011431855327855
3	08139...	Quails	Gecenia	83582 N 55...	Scottsv...	NY	14546	342082376167677
4	40092...	Fairweather	Aaven	47050 E 483...	Satellit...	FL	32937	4409509755979233
5	12758...	Micozzi	Kadel	35443 NE 7...	Sulphu...	TX	75481	6011874713220681

E-R Diagramming

Analyze the structure of, and relationships between, tables you select graphically in any connected database. For more information, see:

www.iri.com/blog/vldb-operations/er-diagrams-in-iri-workbench/

This wizard produces a diagram you can also customize, edit manually to change table attributes, and reproduce in different image formats for display and dissemination.

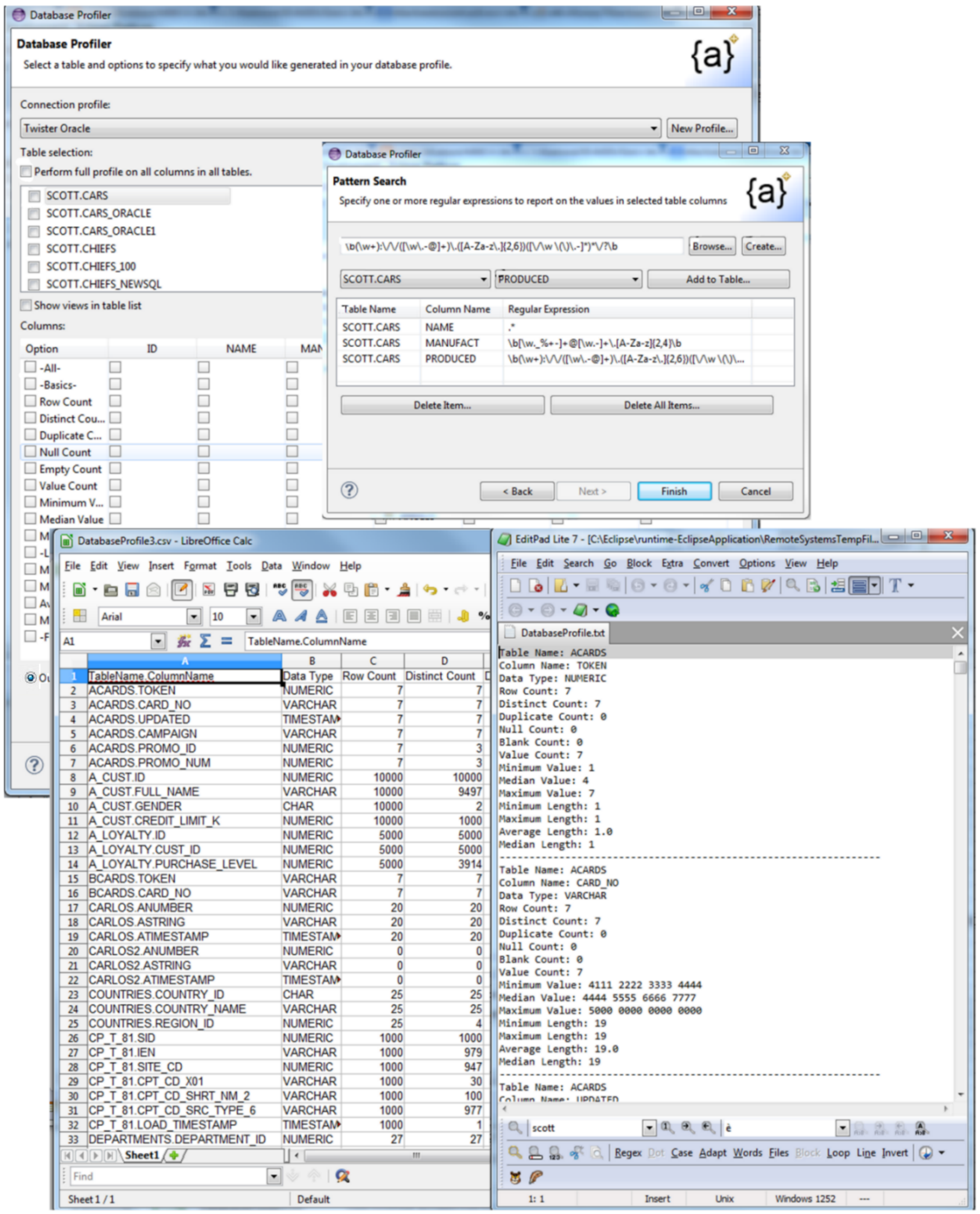


Database & Flat-File Profiling

Through built-in profiling wizards, FieldShield users can produce statistics on the content and character of structured data sources, and search strings in them through matches to data in lookup files, Java RegEx patterns, and fuzzy-matching algorithms.

The data can also check the referential integrity of any (JDBC-connected) relational database.

For more information, see:
www.iri.com/blog/data-transformation2/database-profiling-in-iri-workbench/, and
www.iri.com/blog/iri/iri-workbench/flat-file-profiling/



Database Profiler

Select a table and options to specify what you would like generated in your database profile.

Connection profile: Twister Oracle

Table selection:

- Perform full profile on all columns in all tables.
- SCOTT.CARS
- SCOTT.CARS_ORACLE
- SCOTT.CARS_ORACLE1
- SCOTT.CHIEFS
- SCOTT.CHIEFS_100
- SCOTT.CHIEFS_NEWSQL
- Show views in table list

Columns:

Option	ID	NAME	MAN
-All-			
-Basics-			
Row Count			
Distinct Cou...			
Duplicate C...			
Null Count			
Empty Count			
Value Count			
Minimum V...			
Median Value			

Pattern Search

Specify one or more regular expressions to report on the values in selected table columns

```
\b(\w+):\/\/([\w\.-@]+)\.([A-Za-z\.]{2,6})([\/\w \(\)\.-]*)*\/?\b
```

SCOTT.CARS · PRODUCED

Table Name	Column Name	Regular Expression
SCOTT.CARS	NAME	.*
SCOTT.CARS	MANUFACT	\b[\w._%+-]+@[\w.-]+\.[A-Za-z]{2,4}\b
SCOTT.CARS	PRODUCED	\b(\w+):\/\/([\w\.-@]+)\.([A-Za-z\.]{2,6})([\/\w \(\)\...

DatabaseProfile3.csv - LibreOffice Calc

	A	B	C	D
1	TableName.ColumnName	Data Type	Row Count	Distinct Count
2	ACARDS.TOKEN	NUMERIC	7	7
3	ACARDS.CARD_NO	VARCHAR	7	7
4	ACARDS.UPDATED	TIMESTAM	7	7
5	ACARDS.CAMPAIGN	VARCHAR	7	7
6	ACARDS.PROMO_ID	NUMERIC	7	3
7	ACARDS.PROMO_NUM	NUMERIC	7	3
8	A_CUST.ID	NUMERIC	10000	10000
9	A_CUST.FULL_NAME	VARCHAR	10000	9497
10	A_CUST.GENDER	CHAR	10000	2
11	A_CUST.CREDIT_LIMIT_K	NUMERIC	10000	1000
12	A_LOYALTY.ID	NUMERIC	5000	5000
13	A_LOYALTY.CUST_ID	NUMERIC	5000	5000
14	A_LOYALTY.PURCHASE_LEVEL	NUMERIC	5000	3914
15	BCARDS.TOKEN	VARCHAR	7	7
16	BCARDS.CARD_NO	VARCHAR	7	7
17	CARLOS.ANUMBER	NUMERIC	20	20
18	CARLOS.ASTRING	VARCHAR	20	20
19	CARLOS.ATIMESTAMP	TIMESTAM	20	20
20	CARLOS2.ANUMBER	NUMERIC	0	0
21	CARLOS2.ASTRING	VARCHAR	0	0
22	CARLOS2.ATIMESTAMP	TIMESTAM	0	0
23	COUNTRIES.COUNTRY_ID	CHAR	25	25
24	COUNTRIES.COUNTRY_NAME	VARCHAR	25	25
25	COUNTRIES.REGION_ID	NUMERIC	25	4
26	CP_T_81.SID	NUMERIC	1000	1000
27	CP_T_81.IEN	VARCHAR	1000	979
28	CP_T_81.SITE_CD	NUMERIC	1000	947
29	CP_T_81.CPT_CD_X01	VARCHAR	1000	30
30	CP_T_81.CPT_CD_SHRT_NM_2	VARCHAR	1000	100
31	CP_T_81.CPT_CD_SRC_TYPE_6	VARCHAR	1000	977
32	CP_T_81.LOAD_TIMESTAMP	TIMESTAM	1000	1
33	DEPARTMENTS.DEPARTMENT_ID	NUMERIC	27	27

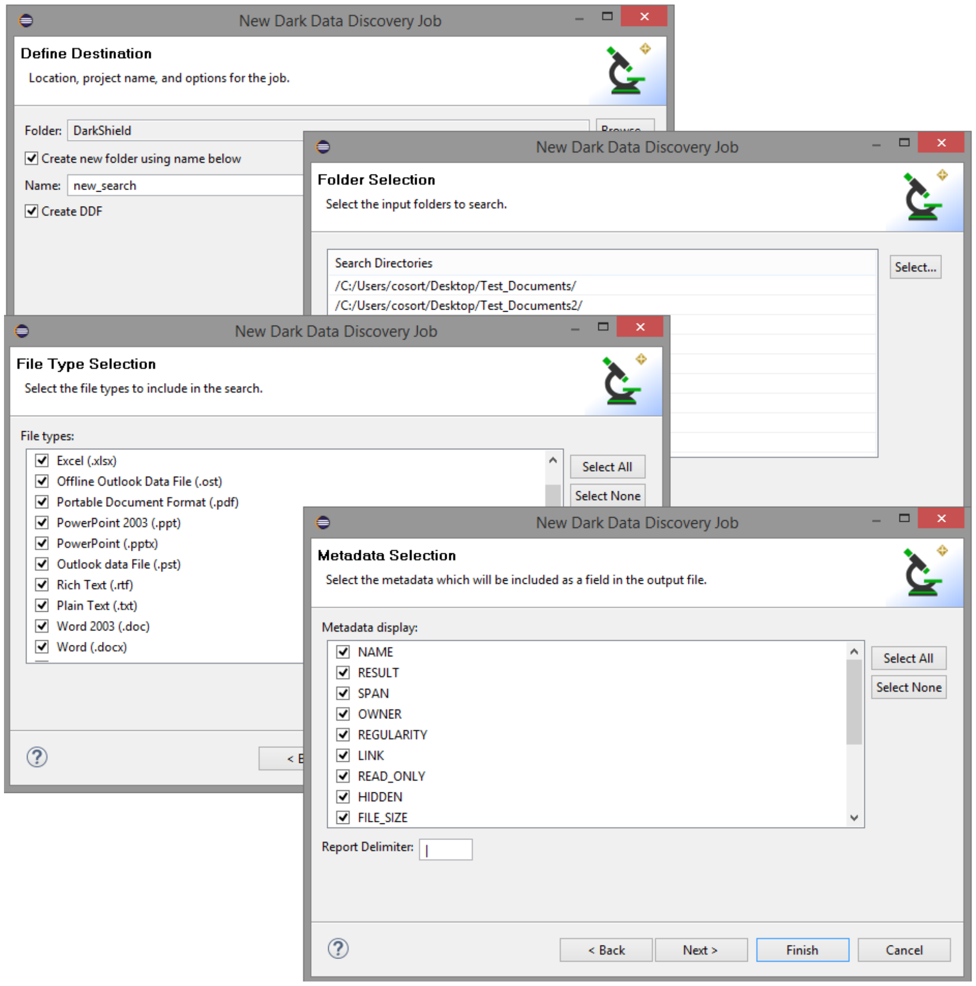
EditPad Lite 7 - [C:\Eclipse\runtime-EclipseApplication\RemoteSystemsTempFil...

DatabaseProfile.txt

```
Table Name: ACARDS
Column Name: TOKEN
Data Type: NUMERIC
Row Count: 7
Distinct Count: 7
Duplicate Count: 0
Null Count: 0
Blank Count: 0
Value Count: 7
Minimum Value: 1
Median Value: 4
Maximum Value: 7
Minimum Length: 1
Maximum Length: 1
Average Length: 1.0
Median Length: 1
----------------------------------------------------------------
Table Name: ACARDS
Column Name: CARD_NO
Data Type: VARCHAR
Row Count: 7
Distinct Count: 7
Duplicate Count: 0
Null Count: 0
Blank Count: 0
Value Count: 7
Minimum Value: 4111 2222 3333 4444
Median Value: 4444 5555 6666 7777
Maximum Value: 5000 0000 0000 0000
Minimum Length: 19
Maximum Length: 19
Average Length: 19.0
Median Length: 19
----------------------------------------------------------------
Table Name: ACARDS
Column Name: UPDATED
```

Unstructured Data Discovery

Because PII also exists in MS Office documents, .pdf, .rtf, and .xml files, logs and images, and other so called “dark data” repositories, it may be important to find, if not mask it. You can use the Dark Data Discovery wizard in IRI Workbench to define one or more patterns to search, and extract the values to DDF-defined flat files, along with forensic metadata about the sources discovered.



Define Destination
Location, project name, and options for the job.

Folder: DarkShield
☑ Create new folder using name below
Name: new_search
☑ Create DDF

Folder Selection
Select the input folders to search.

Search Directories
/C:/Users/cosort/Desktop/Test_Documents/
/C:/Users/cosort/Desktop/Test_Documents2/

File Type Selection
Select the file types to include in the search.

File types:
☑ Excel (.xlsx)
☑ Offline Outlook Data File (.ost)
☑ Portable Document Format (.pdf)
☑ PowerPoint 2003 (.ppt)
☑ PowerPoint (.pptx)
☑ Outlook data File (.pst)
☑ Rich Text (.rtf)
☑ Plain Text (.txt)
☑ Word 2003 (.doc)
☑ Word (.docx)

Metadata Selection
Select the metadata which will be included as a field in the output file.

Metadata display:
☑ NAME
☑ RESULT
☑ SPAN
☑ OWNER
☑ REGULARITY
☑ LINK
☑ READ_ONLY
☑ HIDDEN
☑ FILE_SIZE

Report Delimiter: |

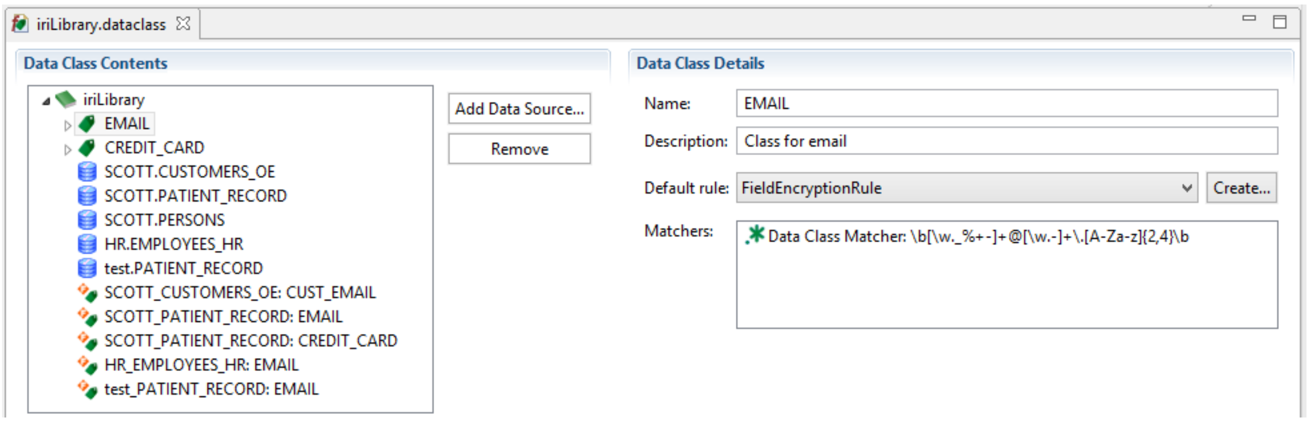
IRI DarkShield can be used for masking data in all of these formats, plus in database BLOB and CLOB columns where PII may be floating within unstructured document, text or image files. For database values exported to Excel, use either DarkShield, or the purpose-built spreadsheet-side masking tool, IRI CellShield EE.

Both DarkShield and CellShield EE can discover and de-identify data using the same data classes, and several of the same masking functions, used in FieldShield. They also share the same dark data discovery wizard for finding PII in multiple files and sheets simultaneously.

Structured Data Discovery & Definition

IRI Workbench wizards connect to flat files and RDBs to find PII in multiple folders and schemas at once and facilitate the application of consistent masking rules. Searches can be performed within the data profiling wizards linked above, or from within fit-for-purpose wide-scan wizards.

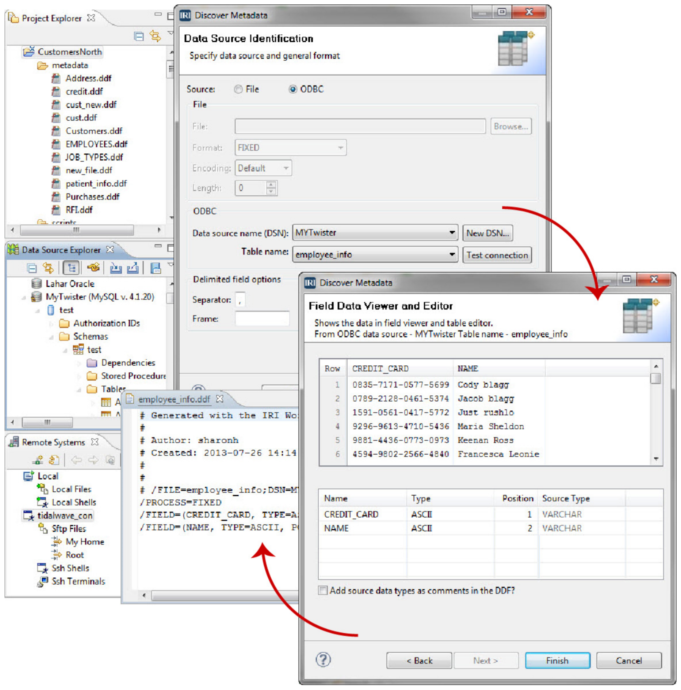
Schema-wide pattern searches can identify columns matching supplied or custom regular expressions. Those results can be associated with data classes. In addition, schema-wide data class searches can find one or more data classes or groups which can activate multiple search methods at once. There is also a wizard for bulk masking across schema and DBs to save time and preserve referential integrity.



iriLibrary
- EMAIL
- CREDIT_CARD
- SCOTT.CUSTOMERS_OE
- SCOTT.PATIENT_RECORD
- SCOTT.PERSONS
- HR.EMPLOYEES_HR
- test.PATIENT_RECORD
- SCOTT_CUSTOMERS_OE: CUST_EMAIL
- SCOTT_PATIENT_RECORD: EMAIL
- SCOTT_PATIENT_RECORD: CREDIT_CARD
- HR_EMPLOYEES_HR: EMAIL
- test_PATIENT_RECORD: EMAIL

Name: EMAIL
Description: Class for email
Default rule: FieldEncryptionRule
Matchers: Data Class Matcher: \b[\w._%+-]+@[\w.-]+\.[A-Za-z]{2,4}\b

FieldShield users can also define and redefine column names, offsets, and data types automatically. They can also save and share the metadata in centralized data definition files and Git repositories. See: www.iri.com/blog/vldb-operations/table-filtering-iri-workbench
www.iri.com/blog/iri/iri-workbench/using-the-metadata-discovery-wizard
www.iri.com/blog/iri/iri-workbench/introduction-metadata-management-hub



Data Source Identification
Specify data source and general format

Source: File [✓ ODBC]
File
File:
Format: FIXED
Encoding: Default
Length: 0
ODBC
Data source name (DSN): MYTwister
Table name: employee_info
Delimited field options
Separator: ,
Frame:

```
# Generated with the IRI Wo
#
# Author: sharonh
# Created: 2013-07-26 14:14
#
#
# /FILE=employee_info;DSN=M
/PROCESS=FIXED
/FIELD=(CREDIT_CARD, TYPE=A
/FIELD=(NAME, TYPE=ASCII, P
```

Field Data Viewer and Editor
Shows the data in field viewer and table editor.
From ODBC data source - MYTwister Table name - employee_info

Row	CREDIT_CARD	NAME
1	0835-7171-0577-5699	Cody blagg
2	0789-2128-0461-5374	Jacob blagg
3	1591-0561-0417-5772	Just rushlo
4	9296-9613-4710-5436	Maria Sheldon
5	9881-4436-0773-0973	Keenan Ross
6	4594-9802-2566-4840	Francesca Leonie

Name	Type	Position	Source Type
CREDIT_CARD	ASCII	1	VARCHAR
NAME	ASCII	2	VARCHAR

☐ Add source data types as comments in the DDF?

Static Data Masking Functions

The protection method you choose for each field depends on your need for security, reversibility, appearance, and speed. In addition to FieldShield's built-in functions, you can also write and call your own field functions at runtime.

FieldShield supports multiple masking functions, in multiple categories:

- multiple, NSA Suite B and FIPS-compliant encryption (and decryption) algorithms, including *format-preserving* encryption
- SHA-1 and SHA-2 hashing
- ASCII de-ID (bit scrambling)
- binary encoding and decoding
- blurring and bucketing (anonymization for HIPAA)
- random value generation or selection
- redaction (full or partial string masking)
- reversible and non-reversible pseudonymization
- filtering or omission (erasure for GDPR)
- literal or lookup value replacement
- byte shifting and (sub)string functions
- tokenization (for PCI DSS compliance)
- custom (user written and linked) routines

FieldShield uses state-of-the-art encryption algorithms that are very difficult to crack. And you can further strengthen encryption with random tokenization, hashing, and secure, or by changing encryption keys. Three encryption key management techniques are built in, and FieldShield is also compatible with the Microsoft Azure Key Vault and Alliance Key Manager from Townsend Security.

FieldShield data obfuscation techniques can also forever protect original values. A non-reversible method like redaction or filtering removes any computational basis for deriving the original value. You can also supply and link your own field-level protection functions for use at runtime with the others.

Masking Job Wizards

Several automatic data masking job creation wizards are supplied in IRI Workbench. There are data class masking wizards for both tables in database schema and structured files in directories that leverage central data definitions and their associated search methods and masking rules. There are also older and simpler wizards that do not require data classification:

One works with a single data source, or multiple sources that share common fields, while the other wizard works with multiple tables with the same or different metadata, and allows you to apply cross-table protection rules to preserve referential integrity after protection occurs. Both wizards use the *New Field Rule Wizard*'s Selection page to help you create or use a saved protection rule.

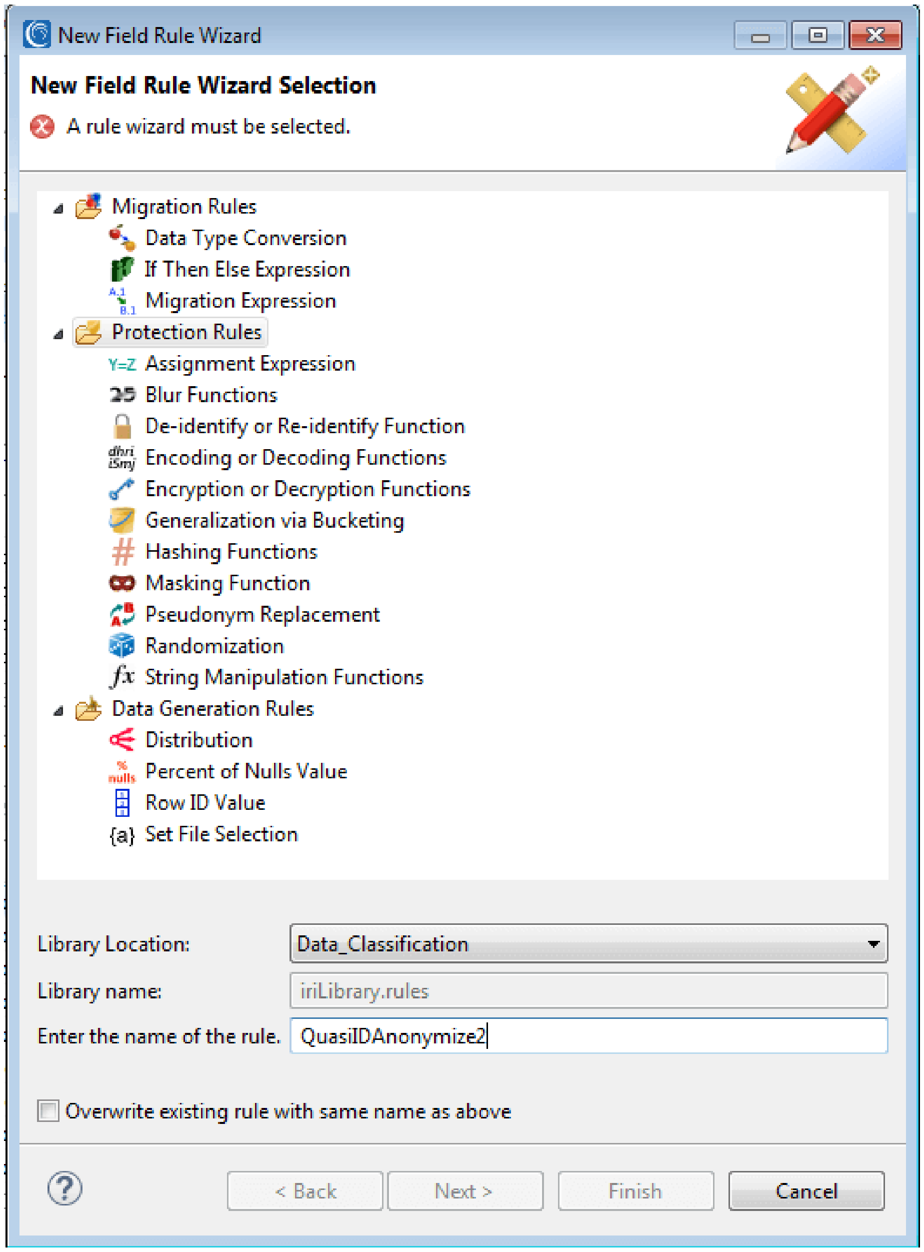
Masking Rules

In the IRI Workbench GUI for FieldShield, built on Eclipse, a variety of protection functions are available through the use of rule wizards that enable you to create simple or complex protection formatting for the script.

You can use the rule one time, or save it to a rule library for multiple or future use. The library is stored in the individual project. You can also create a rule library in each project, or create a rules project in which to store a single library for shared use.

When creating a new protection job with multiple tables and columns, use the Field Modification Rules dialog to create one of the many available types of protection.

When editing an existing job script, use the graphical Target Field Layout editor to create and add rules to the rule library, and then apply them to the job.



New Field Rule Wizard

New Field Rule Wizard Selection

A rule wizard must be selected.

- Migration Rules
  - Data Type Conversion
  - If Then Else Expression
  - Migration Expression
- Protection Rules
  - Assignment Expression
  - Blur Functions
  - De-identify or Re-identify Function
  - Encoding or Decoding Functions
  - Encryption or Decryption Functions
  - Generalization via Bucketing
  - Hashing Functions
  - Masking Function
  - Pseudonym Replacement
  - Randomization
  - String Manipulation Functions
- Data Generation Rules
  - Distribution
  - Percent of Nulls Value
  - Row ID Value
  - Set File Selection

Library Location: Data_Classification

Library name: iriLibrary.rules

Enter the name of the rule. QuasiIDAnonymize2

☐ Overwrite existing rule with same name as above

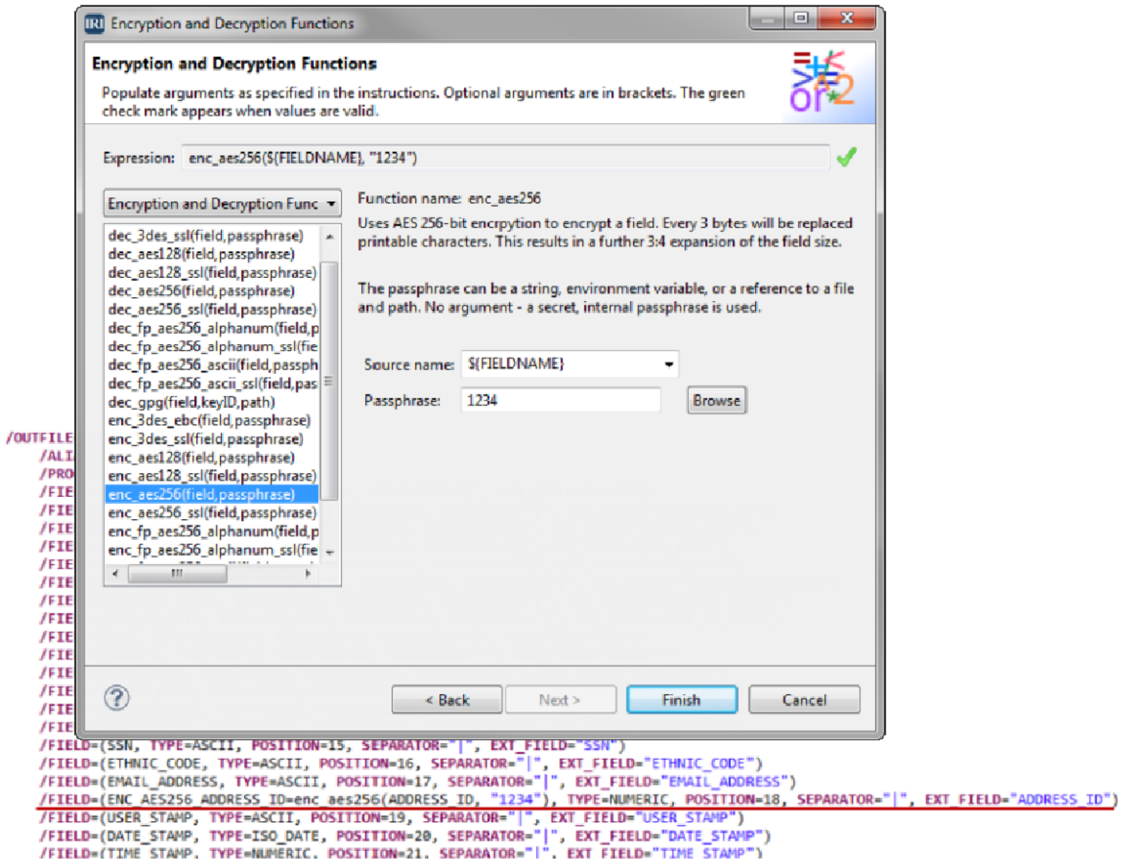
Masking Function Dialogs

Most FieldShield functions can be applied visually in IRI Workbench dialogs, which appear in new job wizards, or when opened from the job script window or outline, or visual workflow.

Encryption and Decryption

Encryption is one of the best ways to protect PII and primary account numbers (PANs) and other sensitive data because you use a specific key that makes the encrypted data generally unreadable. The data can only be decrypted using a matching decryption protocol and key.

FieldShield includes 3DES, AES, FIPS-compliant OpenSSL, and GPG encryption and decryption libraries. That dialog is shown below, but remember that you can also supply your own, custom encryption and decryption functions that you write and link.



Encryption and Decryption Functions

Encryption and Decryption Functions

Populate arguments as specified in the instructions. Optional arguments are in brackets. The green check mark appears when values are valid.

Expression: enc_aes256(${FIELDNAME}, "1234") ✓

Encryption and Decryption Func

```
dec_3des_ssl(field,passphrase)
dec_aes128(field,passphrase)
dec_aes128_ssl(field,passphrase)
dec_aes256(field,passphrase)
dec_aes256_ssl(field,passphrase)
dec_fp_aes256_alphanum(field,p
dec_fp_aes256_alphanum_ssl(fie
dec_fp_aes256_ascii(field,passph
dec_fp_aes256_ascii_ssl(field,pas
dec_gpg(field,keyID,path)
enc_3des_ebc(field,passphrase)
enc_3des_ssl(field,passphrase)
enc_aes128(field,passphrase)
enc_aes128_ssl(field,passphrase)
enc_aes256(field,passphrase)
enc_aes256_ssl(field,passphrase)
enc_fp_aes256_alphanum(field,p
enc_fp_aes256_alphanum_ssl(fie
```

Function name: enc_aes256

Uses AES 256-bit encrpytion to encrypt a field. Every 3 bytes will be replaced printable characters. This results in a further 3:4 expansion of the field size.

The passphrase can be a string, environment variable, or a reference to a file and path. No argument - a secret, internal passphrase is used.

Source name: ${FIELDNAME}

Passphrase: 1234

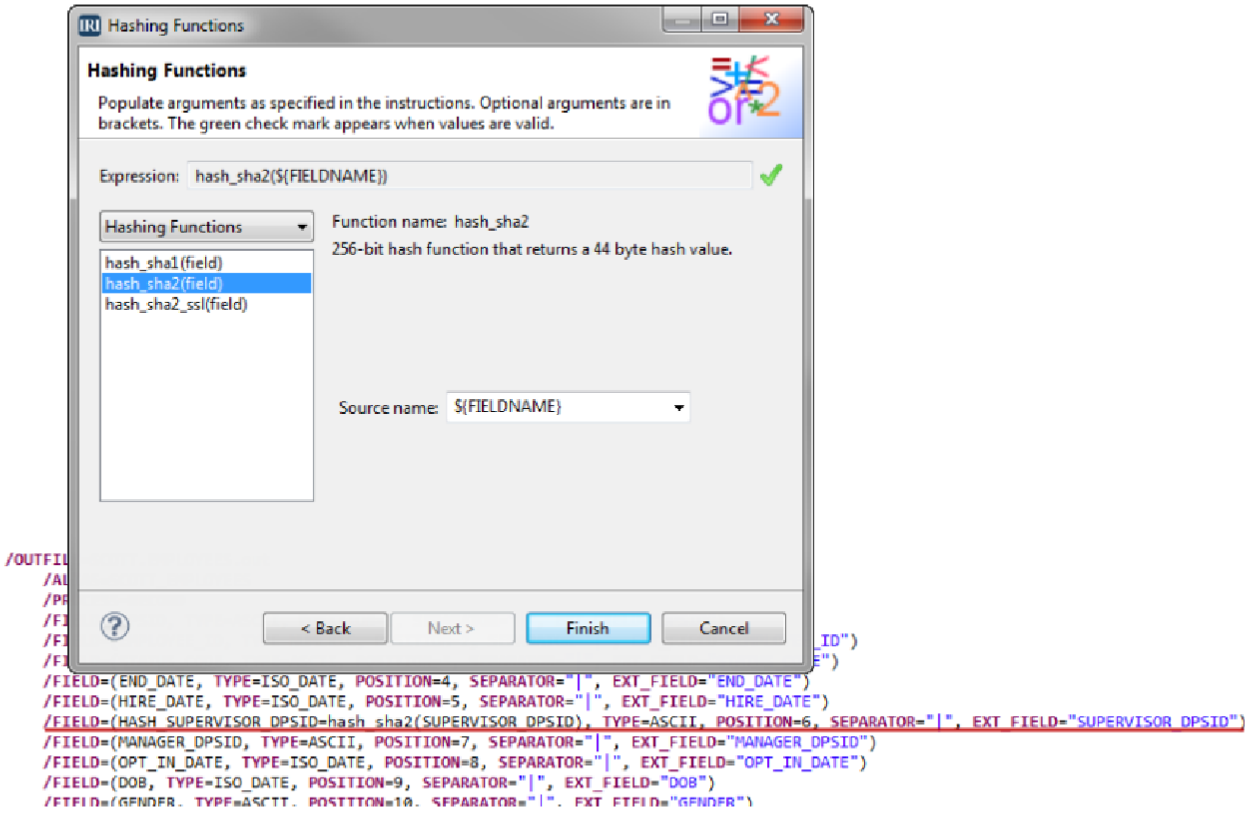
```
/OUTFILE
   /ALI
   /PRO
   /FIE
   /FIE
   /FIE
   /FIE
   /FIE
   /FIE
   /FIE
   /FIE
   /FIE
   /FIE
   /FIE
   /FIE
   /FIE
   /FIE
   /FIELD=(SSN, TYPE=ASCII, POSITION=15, SEPARATOR="|", EXT_FIELD="SSN")
   /FIELD=(ETHNIC_CODE, TYPE=ASCII, POSITION=16, SEPARATOR="|", EXT_FIELD="ETHNIC_CODE")
   /FIELD=(EMAIL_ADDRESS, TYPE=ASCII, POSITION=17, SEPARATOR="|", EXT_FIELD="EMAIL_ADDRESS")
   /FIELD=(ENC_AES256_ADDRESS_ID=enc_aes256(ADDRESS_ID, "1234"), TYPE=NUMERIC, POSITION=18, SEPARATOR="|", EXT_FIELD="ADDRESS_ID")
   /FIELD=(USER_STAMP, TYPE=ASCII, POSITION=19, SEPARATOR="|", EXT_FIELD="USER_STAMP")
   /FIELD=(DATE_STAMP, TYPE=ISO_DATE, POSITION=20, SEPARATOR="|", EXT_FIELD="DATE_STAMP")
   /FIELD=(TIME_STAMP, TYPE=NUMERIC, POSITION=21, SEPARATOR="|", EXT_FIELD="TIME_STAMP")
```

The most popular FieldShield function is 256-bit AES format-preserving encryption because the resulting ciphertext resembles the original field format and value.

FieldShield supports multiple encryption key management methods as well, to help authenticate users, and secure and rotate keys.

Hashing

Hashing is a difficult-to-reverse data masking technique that converts a variable length "message" (e.g., someone's password) into an obfuscated, fixed-length, alphanumeric string. The purpose of the (Secure Hash Algorithm) SHA-2 field-level routine, for example, is to return a hash of the data string in a given field or column.



Populate arguments as specified in the instructions. Optional arguments are in brackets. The green check mark appears when values are valid.

Expression: hash_sha2(${FIELDNAME})

Hashing Functions

hash_sha1(field)
hash_sha2(field)
hash_sha2_ssl(field)

Function name: hash_sha2
256-bit hash function that returns a 44 byte hash value.

Source name: ${FIELDNAME}

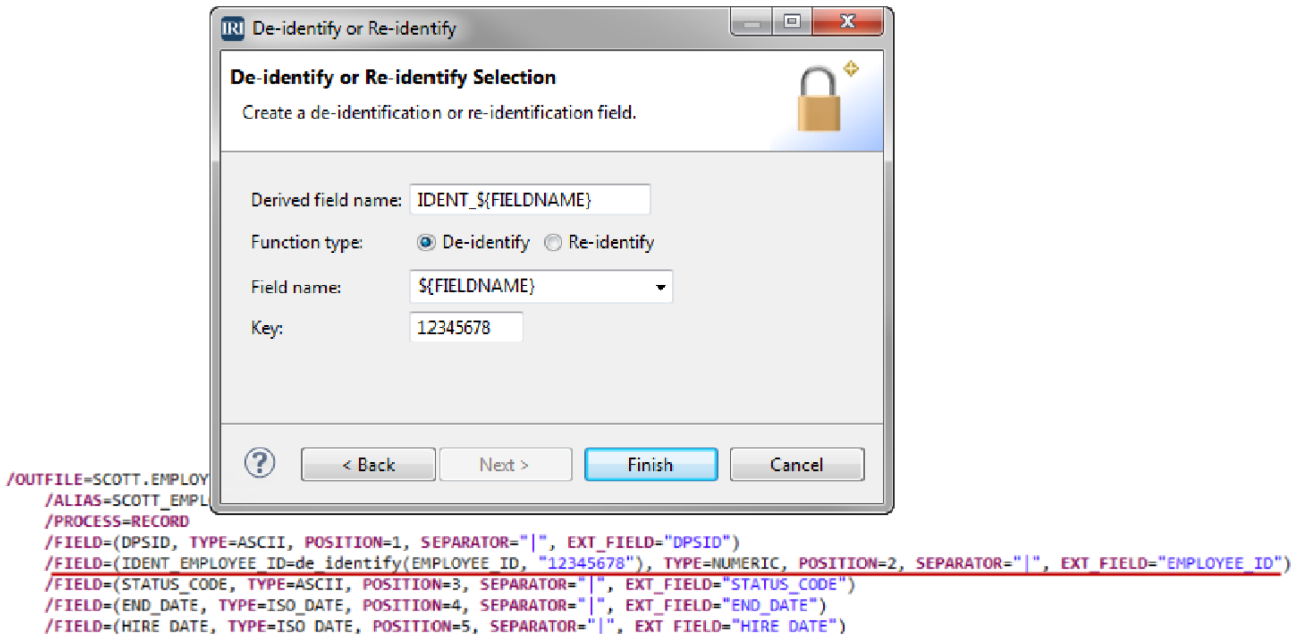
```
/OUTFIL
    /AL
    /PR
    /FI
    /FI                                                                                  ID")
    /FI                                                                                  E")
    /FIELD=(END_DATE, TYPE=ISO_DATE, POSITION=4, SEPARATOR="|", EXT_FIELD="END_DATE")
    /FIELD=(HIRE_DATE, TYPE=ISO_DATE, POSITION=5, SEPARATOR="|", EXT_FIELD="HIRE_DATE")
    /FIELD=(HASH_SUPERVISOR_DPSID=hash_sha2(SUPERVISOR_DPSID), TYPE=ASCII, POSITION=6, SEPARATOR="|", EXT_FIELD="SUPERVISOR_DPSID")
    /FIELD=(MANAGER_DPSID, TYPE=ASCII, POSITION=7, SEPARATOR="|", EXT_FIELD="MANAGER_DPSID")
    /FIELD=(OPT_IN_DATE, TYPE=ISO_DATE, POSITION=8, SEPARATOR="|", EXT_FIELD="OPT_IN_DATE")
    /FIELD=(DOB, TYPE=ISO_DATE, POSITION=9, SEPARATOR="|", EXT_FIELD="DOB")
    /FIELD=(GENDER, TYPE=ASCII, POSITION=10, SEPARATOR="|", EXT_FIELD="GENDER")
```

This is also known as a cryptographic hashing function. For any number of input bytes, the routine always returns a 32-byte hash code (in binary) that internally translates to a 44-byte, base64 value. Hashing is a useful technique for integrity checking. SHA-1, SHA-2, and SHA-2 OpenSSL are available.

Bit Scrambling (ASCII De/Re-ID)

FieldShield can de- and re-identify PII, protected health information (PHI), and other sensitive field data in databases and files by scrambling data. This uses less computational overhead (and security) than encryption and decryption. At the same time, the de-identified field can look more realistic and still be recovered by using an encryption-style key.

The de-identification key used by FieldShield is a user-specified string, where the same string is used for subsequent re-identification. The hexadecimal range of the data contents to be de-identified must fall within the ASCII printable character range.


De-identify or Re-identify

De-identify or Re-identify Selection

Create a de-identification or re-identification field.

Derived field name: IDENT_${FIELDNAME}

Function type: [✓ De-identify] Re-identify

Field name: ${FIELDNAME}

Key: 12345678

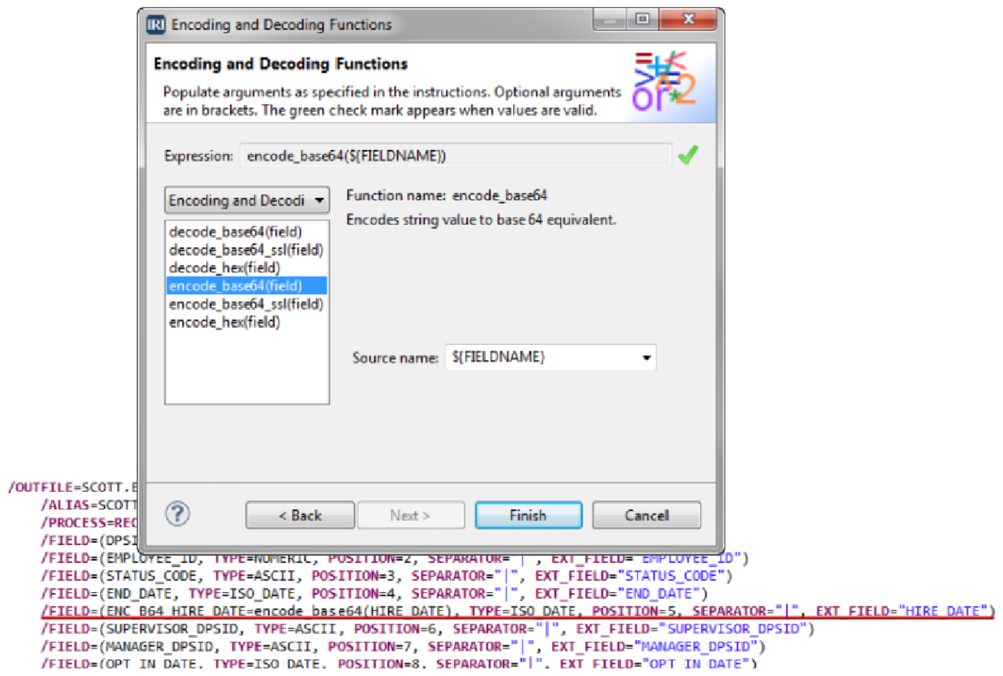
```
/OUTFILE=SCOTT.EMPLOY
    /ALIAS=SCOTT_EMPL
    /PROCESS=RECORD
    /FIELD=(DPSID, TYPE=ASCII, POSITION=1, SEPARATOR="|", EXT_FIELD="DPSID")
    /FIELD=(IDENT_EMPLOYEE_ID=de_identify(EMPLOYEE_ID, "12345678"), TYPE=NUMERIC, POSITION=2, SEPARATOR="|", EXT_FIELD="EMPLOYEE_ID")
    /FIELD=(STATUS_CODE, TYPE=ASCII, POSITION=3, SEPARATOR="|", EXT_FIELD="STATUS_CODE")
    /FIELD=(END_DATE, TYPE=ISO_DATE, POSITION=4, SEPARATOR="|", EXT_FIELD="END_DATE")
    /FIELD=(HIRE_DATE, TYPE=ISO_DATE, POSITION=5, SEPARATOR="|", EXT_FIELD="HIRE_DATE")
```

Encoding and Decoding

Unlike encryption functions FieldShield supplies, encoding is not a particularly secure method of de-identification. While it does not require a passphrase or key to reverse, it is convenient and secure enough for some applications. Only the equivalent decoding function provided in FieldShield will reveal the original source values.

Use encoding routines when you need to encode byte data to be stored and transferred over media that are designed to deal with text data, such as email.

Three types of encoding convert the data in an ASCII string into its base64 equivalent value. An OpenSSL implementation of base 64 encoding is also available. The third type of encoding returns the hexadecimal representation of the byte input.


Encoding and Decoding Functions

Encoding and Decoding Functions

Populate arguments as specified in the instructions. Optional arguments are in brackets. The green check mark appears when values are valid.

Expression: encode_base64(${FIELDNAME}) ✓

Encoding and Decodi

- decode_base64(field)
- decode_base64_ssl(field)
- decode_hex(field)
- encode_base64(field)
- encode_base64_ssl(field)
- encode_hex(field)

Function name: encode_base64

Encodes string value to base 64 equivalent.

Source name: ${FIELDNAME}

```
/OUTFILE=SCOTT.E
    /ALIAS=SCOTT
    /PROCESS=REC
    /FIELD=(DPSI
    /FIELD=(EMPLOYEE_ID, TYPE=NUMERIC, POSITION=2, SEPARATOR="|", EXT_FIELD="EMPLOYEE_ID")
    /FIELD=(STATUS_CODE, TYPE=ASCII, POSITION=3, SEPARATOR="|", EXT_FIELD="STATUS_CODE")
    /FIELD=(END_DATE, TYPE=ISO_DATE, POSITION=4, SEPARATOR="|", EXT_FIELD="END_DATE")
    /FIELD=(ENC_B64_HIRE_DATE=encode_base64(HIRE_DATE), TYPE=ISO_DATE, POSITION=5, SEPARATOR="|", EXT_FIELD="HIRE_DATE")
    /FIELD=(SUPERVISOR_DPSID, TYPE=ASCII, POSITION=6, SEPARATOR="|", EXT_FIELD="SUPERVISOR_DPSID")
    /FIELD=(MANAGER_DPSID, TYPE=ASCII, POSITION=7, SEPARATOR="|", EXT_FIELD="MANAGER_DPSID")
    /FIELD=(OPT_IN_DATE, TYPE=ISO_DATE, POSITION=8, SEPARATOR="|", EXT_FIELD="OPT_IN_DATE")
```

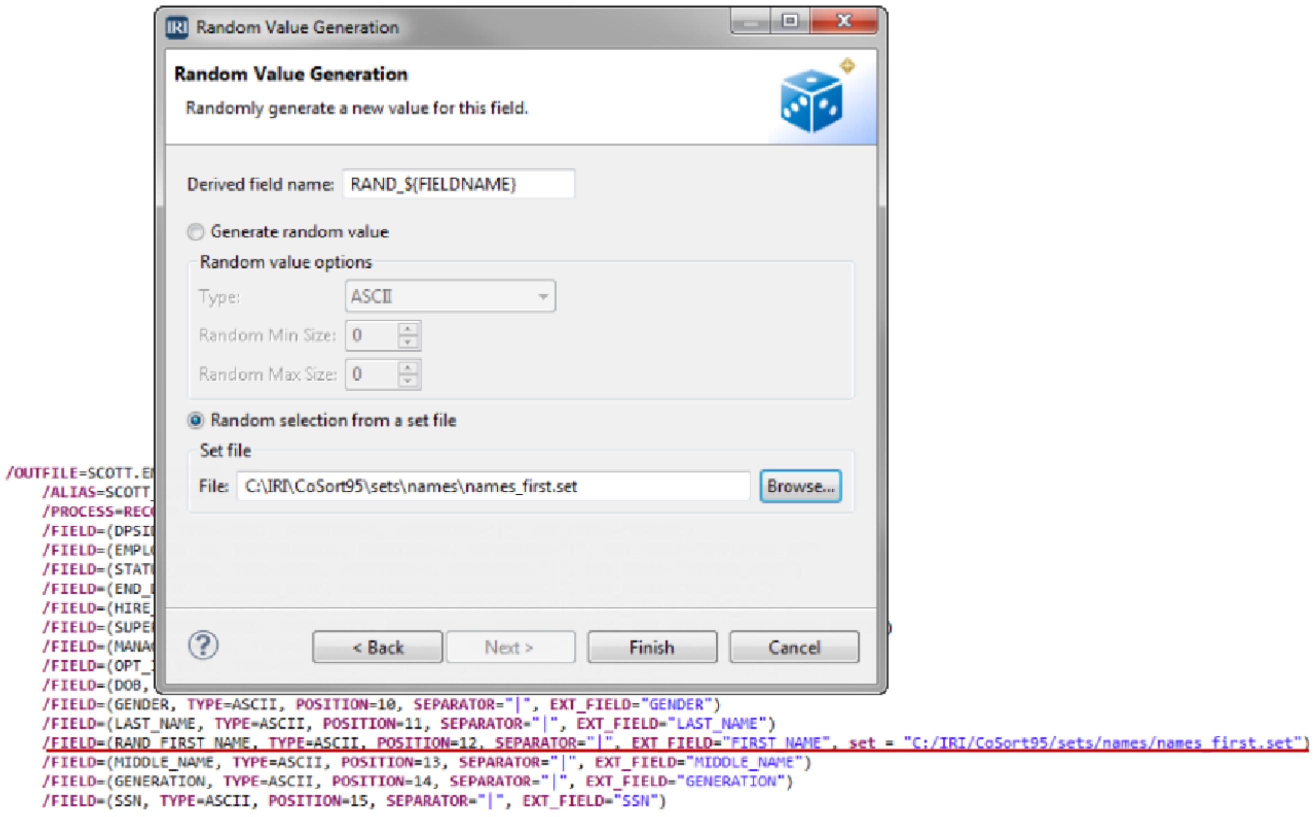
Randomization

Replacing real values with random values is another secure, but non-reversible way to shield PII from disclosure, while maintaining the original structure of the input data. It can be utilized in two ways:

Random data generation, where random values are generated based on a data type with a random minimum and maximum size. Using random data, especially numbers, can make the protected field look somewhat real, while not requiring the use of any real data or protective overhead.

Random selection from a set file, which can use more realistic data from:

- an alpha-numeric list of values; the output values appear as they do in the set file.
- numeric values and ranges; the output values contain only the listed numbers, or a specified range of numbers.
- date values and ranges; only listed dates, or dates within a specific range, appear.



Random Value Generation

Random Value Generation

Randomly generate a new value for this field.

Derived field name: RAND_${FIELDNAME}

( ) Generate random value

Random value options

Type: ASCII

Random Min Size: 0

Random Max Size: 0

(●) Random selection from a set file

Set file

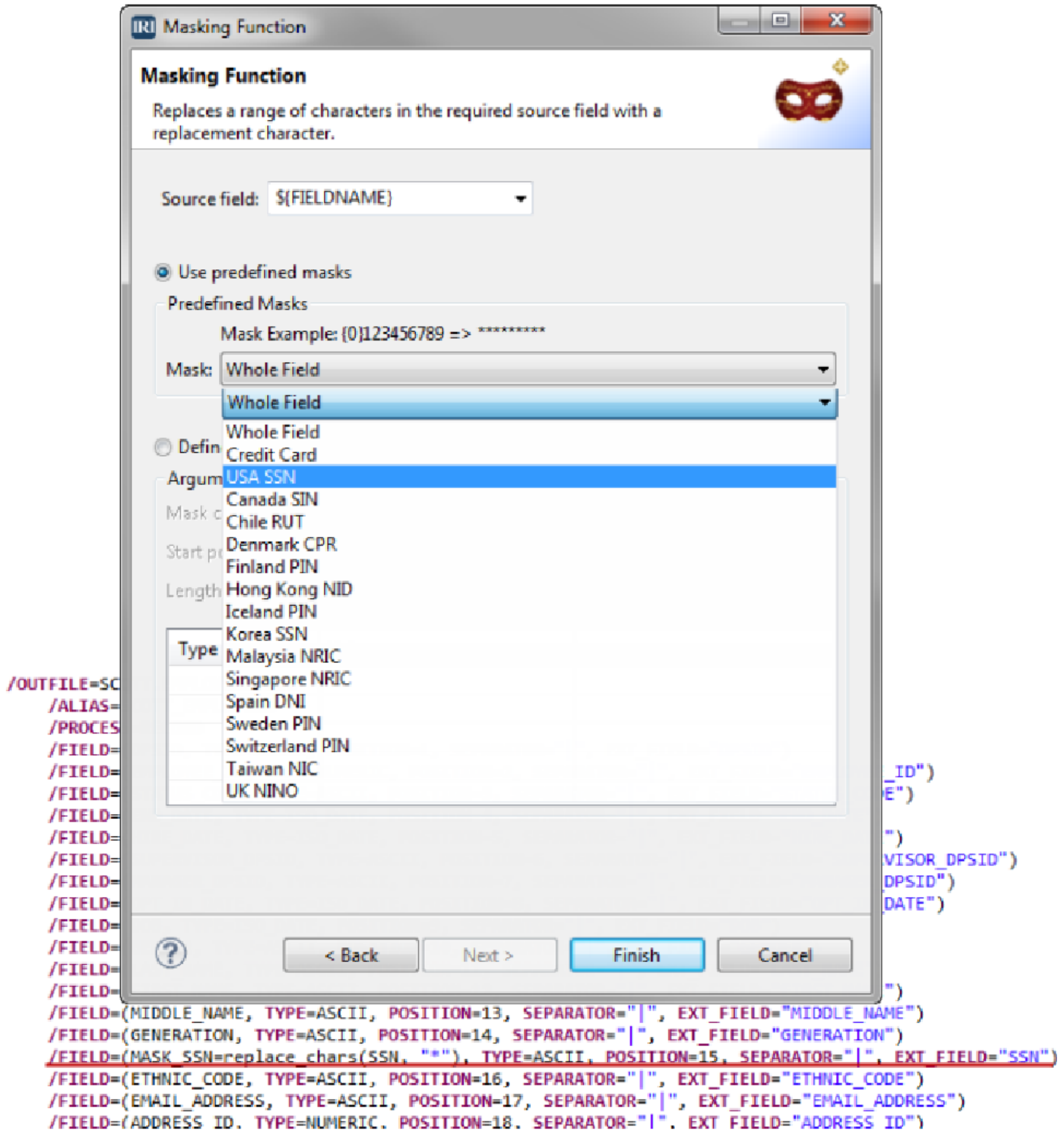
File: C:\IRI\CoSort95\sets\names\names_first.set

```
/OUTFILE=SCOTT.E
    /ALIAS=SCOTT_
    /PROCESS=REC
    /FIELD=(DPSI
    /FIELD=(EMPL
    /FIELD=(STAT
    /FIELD=(END_
    /FIELD=(HIRE
    /FIELD=(SUPE
    /FIELD=(MANA
    /FIELD=(OPT_
    /FIELD=(DOB,
    /FIELD=(GENDER, TYPE=ASCII, POSITION=10, SEPARATOR="|", EXT_FIELD="GENDER")
    /FIELD=(LAST_NAME, TYPE=ASCII, POSITION=11, SEPARATOR="|", EXT_FIELD="LAST_NAME")
    /FIELD=(RAND_FIRST_NAME, TYPE=ASCII, POSITION=12, SEPARATOR="|", EXT_FIELD="FIRST_NAME", set = "C:/IRI/CoSort95/sets/names/names_first.set")
    /FIELD=(MIDDLE_NAME, TYPE=ASCII, POSITION=13, SEPARATOR="|", EXT_FIELD="MIDDLE_NAME")
    /FIELD=(GENERATION, TYPE=ASCII, POSITION=14, SEPARATOR="|", EXT_FIELD="GENERATION")
    /FIELD=(SSN, TYPE=ASCII, POSITION=15, SEPARATOR="|", EXT_FIELD="SSN")
```

Redaction (String Masking)

Personally identifying information (PII) that is not needed for downstream processing or recovery should be redacted (masked) to prevent misuse. While DBAs can remove columns in tables, there are few alternatives for masking data in different ways across multiple databases and flat-file sources.

Only selected parts of the data should be exposed. At the same time, anonymization should preserve the original storage format and overall field appearance so that the platform structure or application need not be altered. Masking all but the last four digits of a credit card or social security number is a common requirement.



Masking Function

Replaces a range of characters in the required source field with a replacement character.

Source field: ${FIELDNAME}

[✓ Use predefined masks]

Predefined Masks

Mask Example: {0}123456789 => *********

Mask: Whole Field

Whole Field
Credit Card
USA SSN
Canada SIN
Chile RUT
Denmark CPR
Finland PIN
Hong Kong NID
Iceland PIN
Korea SSN
Malaysia NRIC
Singapore NRIC
Spain DNI
Sweden PIN
Switzerland PIN
Taiwan NIC
UK NINO

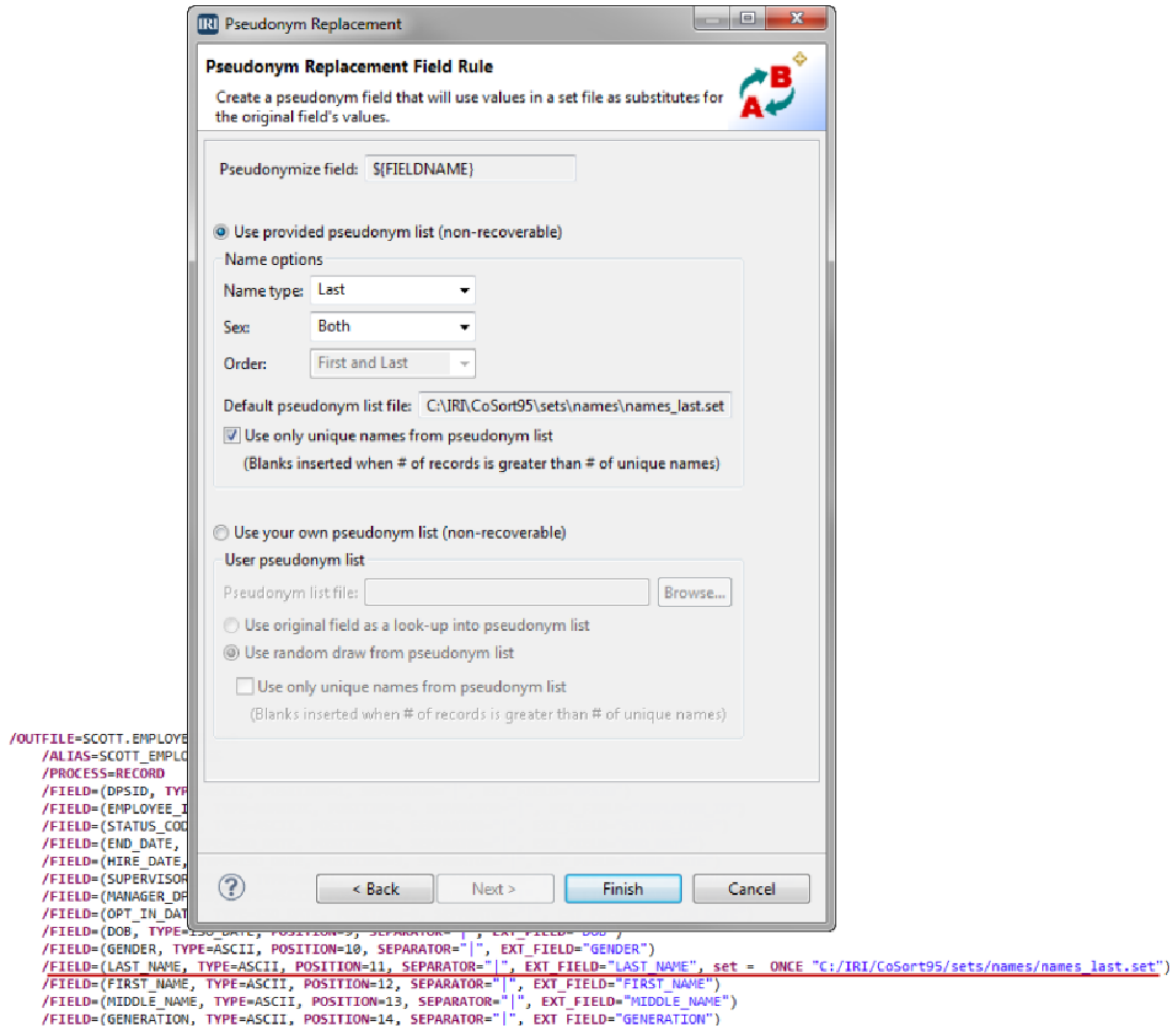
```
/FIELD=(MIDDLE_NAME, TYPE=ASCII, POSITION=13, SEPARATOR="|", EXT_FIELD="MIDDLE_NAME")
/FIELD=(GENERATION, TYPE=ASCII, POSITION=14, SEPARATOR="|", EXT_FIELD="GENERATION")
/FIELD=(MASK_SSN=replace_chars(SSN, "*"), TYPE=ASCII, POSITION=15, SEPARATOR="|", EXT_FIELD="SSN")
/FIELD=(ETHNIC_CODE, TYPE=ASCII, POSITION=16, SEPARATOR="|", EXT_FIELD="ETHNIC_CODE")
/FIELD=(EMAIL_ADDRESS, TYPE=ASCII, POSITION=17, SEPARATOR="|", EXT_FIELD="EMAIL_ADDRESS")
/FIELD=(ADDRESS_ID, TYPE=NUMERIC, POSITION=18, SEPARATOR="|", EXT_FIELD="ADDRESS_ID")
```

Pseudonymization

Pseudonymization provides the level of realism you may require for your business. Two types of pseudonymization are available in FieldShield: recoverable and non-recoverable.

The recoverable function builds a lookup, or two-column set, file containing the original names from the input source, and substitution values randomly selected from that data to be paired with the original values. This effectively creates a shuffled list of real names so that there is no direct association between the original people and their attributes within other columns.

The non-recoverable function performs a random lookup from a supplied list, or single-column set file of names. You can choose among common western first, last or both names, as well as gender. This allows the names to appear real without compromising any one's identity.



Pseudonym Replacement Field Rule

Create a pseudonym field that will use values in a set file as substitutes for the original field's values.

Pseudonymize field: ${FIELDNAME}

● Use provided pseudonym list (non-recoverable)

Name options

Name type: Last

Sex: Both

Order: First and Last

Default pseudonym list file: C:\IRI\CoSort95\sets\names\names_last.set

☑ Use only unique names from pseudonym list

(Blanks inserted when # of records is greater than # of unique names)

○ Use your own pseudonym list (non-recoverable)

User pseudonym list

Pseudonym list file: Browse...

○ Use original field as a look-up into pseudonym list

● Use random draw from pseudonym list

☐ Use only unique names from pseudonym list

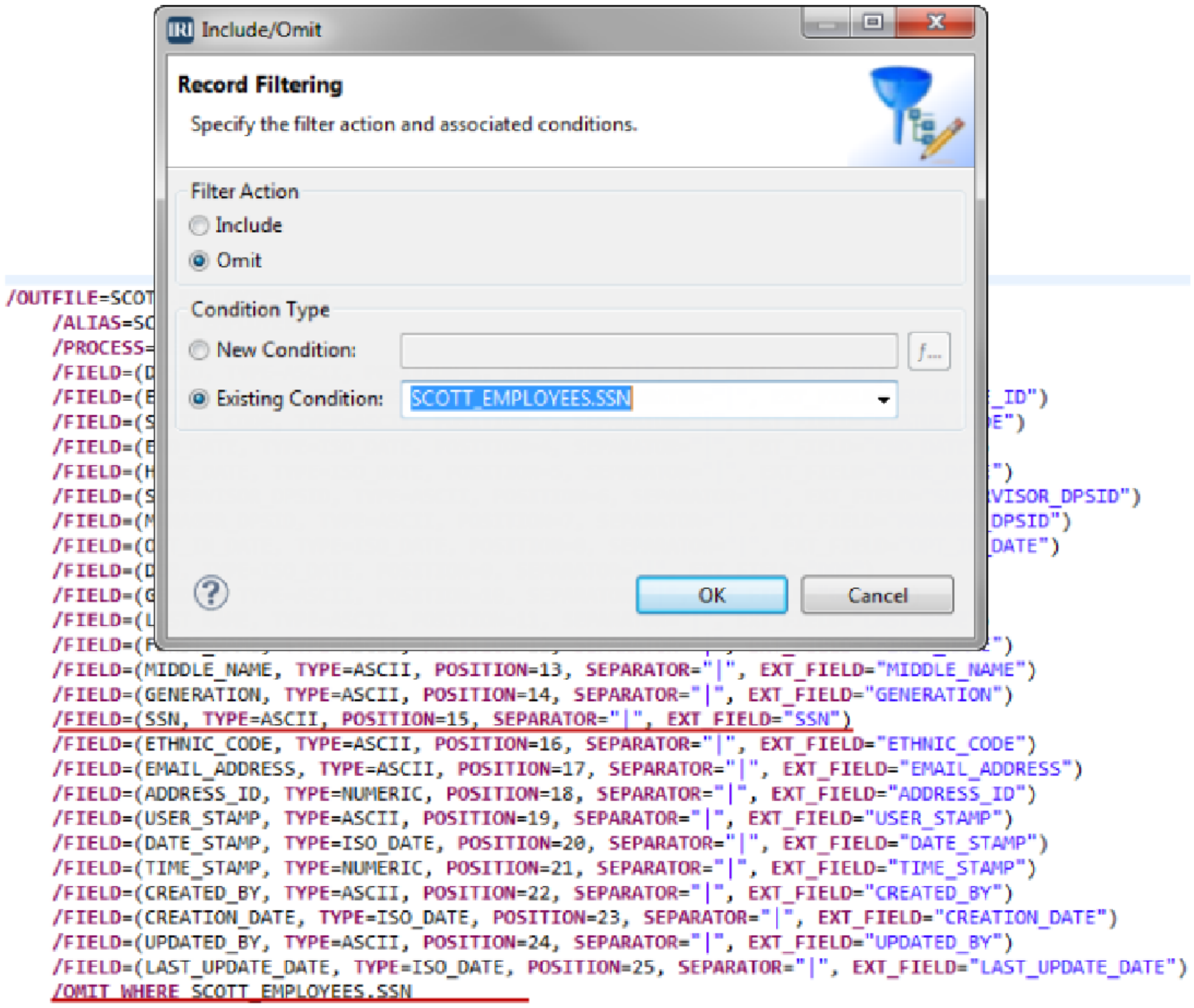
(Blanks inserted when # of records is greater than # of unique names)

```
/OUTFILE=SCOTT.EMPLOYE
    /ALIAS=SCOTT_EMPLO
    /PROCESS=RECORD
    /FIELD=(DPSID, TYP
    /FIELD=(EMPLOYEE_I
    /FIELD=(STATUS_COD
    /FIELD=(END_DATE,
    /FIELD=(HIRE_DATE,
    /FIELD=(SUPERVISOR
    /FIELD=(MANAGER_DP
    /FIELD=(OPT_IN_DAT
    /FIELD=(DOB, TYPE=
    /FIELD=(GENDER, TYPE=ASCII, POSITION=10, SEPARATOR="|", EXT_FIELD="GENDER")
    /FIELD=(LAST_NAME, TYPE=ASCII, POSITION=11, SEPARATOR="|", EXT_FIELD="LAST_NAME", set =  ONCE "C:/IRI/CoSort95/sets/names/names_last.set")
    /FIELD=(FIRST_NAME, TYPE=ASCII, POSITION=12, SEPARATOR="|", EXT_FIELD="FIRST_NAME")
    /FIELD=(MIDDLE_NAME, TYPE=ASCII, POSITION=13, SEPARATOR="|", EXT_FIELD="MIDDLE_NAME")
    /FIELD=(GENERATION, TYPE=ASCII, POSITION=14, SEPARATOR="|", EXT FIELD="GENERATION")
```

IRI has also documented a method for maintaining unique pseudonyms and avoiding collisions as source values grow. See this article.

Filtering or Removal

Removal is useful when sensitive data is not allowed in production or replicated data stores, or when it is just not needed in a future process step. You can easily remove columns or specific values from a database or file.



Include/Omit

Record Filtering

Specify the filter action and associated conditions.

Filter Action

Include [✓ Omit]

Condition Type

New Condition: [✓ Existing Condition: SCOTT_EMPLOYEES.SSN]

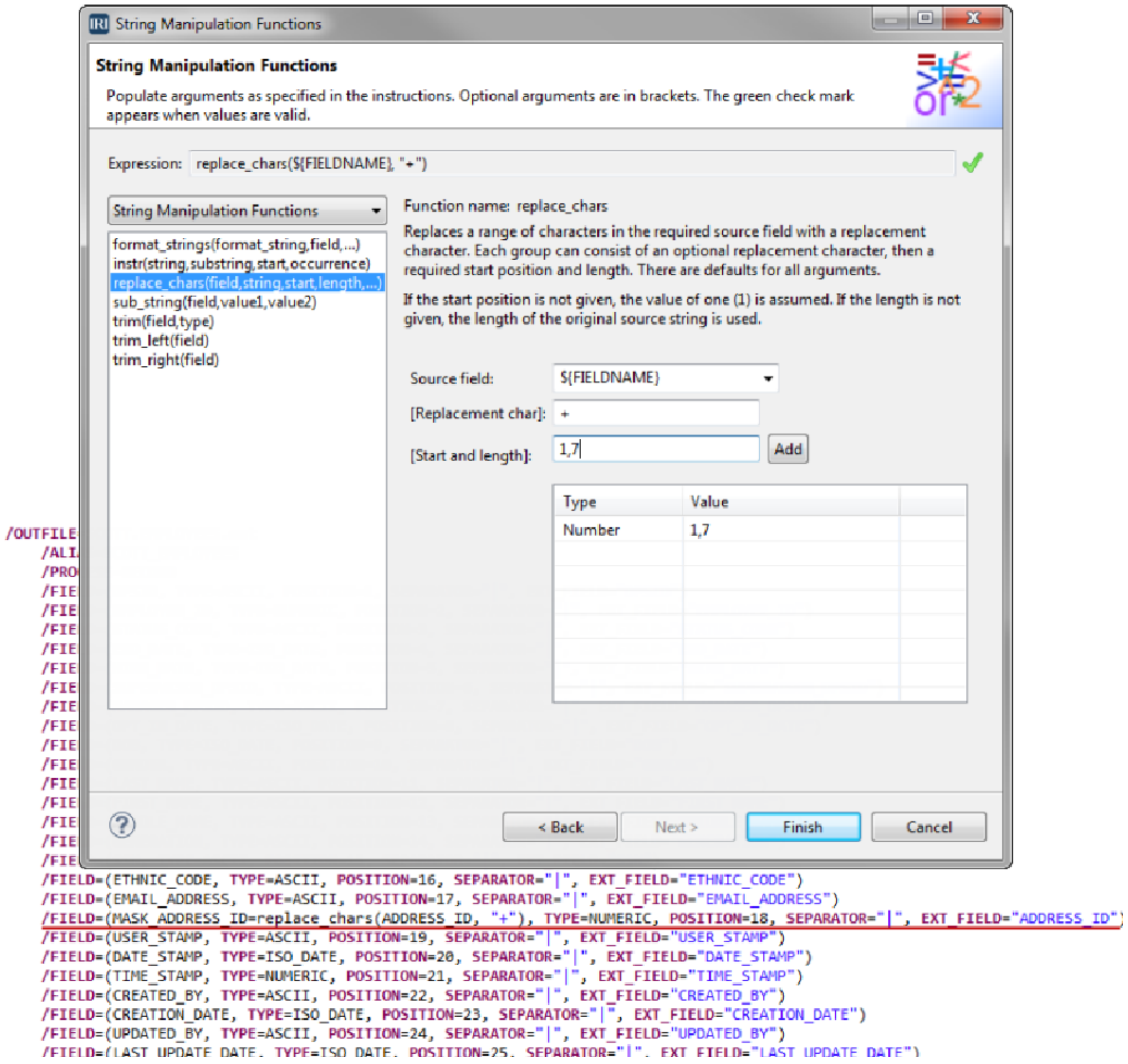
```
/OUTFILE=SCOT
    /ALIAS=SC
    /PROCESS=
    /FIELD=(D
    /FIELD=(E                                                                    _ID")
    /FIELD=(S                                                                    E")
    /FIELD=(E
    /FIELD=(H                                                                    ")
    /FIELD=(S                                                                    VISOR_DPSID")
    /FIELD=(M                                                                    DPSID")
    /FIELD=(O                                                                    DATE")
    /FIELD=(D
    /FIELD=(G
    /FIELD=(L
    /FIELD=(F                                                                    ")
    /FIELD=(MIDDLE_NAME, TYPE=ASCII, POSITION=13, SEPARATOR="|", EXT_FIELD="MIDDLE_NAME")
    /FIELD=(GENERATION, TYPE=ASCII, POSITION=14, SEPARATOR="|", EXT_FIELD="GENERATION")
    /FIELD=(SSN, TYPE=ASCII, POSITION=15, SEPARATOR="|", EXT_FIELD="SSN")
    /FIELD=(ETHNIC_CODE, TYPE=ASCII, POSITION=16, SEPARATOR="|", EXT_FIELD="ETHNIC_CODE")
    /FIELD=(EMAIL_ADDRESS, TYPE=ASCII, POSITION=17, SEPARATOR="|", EXT_FIELD="EMAIL_ADDRESS")
    /FIELD=(ADDRESS_ID, TYPE=NUMERIC, POSITION=18, SEPARATOR="|", EXT_FIELD="ADDRESS_ID")
    /FIELD=(USER_STAMP, TYPE=ASCII, POSITION=19, SEPARATOR="|", EXT_FIELD="USER_STAMP")
    /FIELD=(DATE_STAMP, TYPE=ISO_DATE, POSITION=20, SEPARATOR="|", EXT_FIELD="DATE_STAMP")
    /FIELD=(TIME_STAMP, TYPE=NUMERIC, POSITION=21, SEPARATOR="|", EXT_FIELD="TIME_STAMP")
    /FIELD=(CREATED_BY, TYPE=ASCII, POSITION=22, SEPARATOR="|", EXT_FIELD="CREATED_BY")
    /FIELD=(CREATION_DATE, TYPE=ISO_DATE, POSITION=23, SEPARATOR="|", EXT_FIELD="CREATION_DATE")
    /FIELD=(UPDATED_BY, TYPE=ASCII, POSITION=24, SEPARATOR="|", EXT_FIELD="UPDATED_BY")
    /FIELD=(LAST_UPDATE_DATE, TYPE=ISO_DATE, POSITION=25, SEPARATOR="|", EXT_FIELD="LAST_UPDATE_DATE")
    /OMIT WHERE SCOTT_EMPLOYEES.SSN
```

By removing data in this or other ways, FieldShield users can erase or delete source data in support of “Right to Be Forgotten” provisions of data privacy laws like the GDPR.

Expression and String Functions

Numeric fields can be protected by applying mathematical calculation and trigonometric functions to the data. It is reversible only if you use the same expression for everything.

Bit-level functions can also manipulate column values. Available functions include string “find and replace”, repositioning, trimming, and/or the use of INSTR information. Depending on the function and how you use it, the result of your manipulation may or may not be reversible or compliant.



String Manipulation Functions

String Manipulation Functions

Populate arguments as specified in the instructions. Optional arguments are in brackets. The green check mark appears when values are valid.

Expression: replace_chars(${FIELDNAME}, "+") ✓

String Manipulation Functions

- format_strings(format_string,field,...)
- instr(string,substring,start,occurrence)
- replace_chars(field,string,start,length,...)
- sub_string(field,value1,value2)
- trim(field,type)
- trim_left(field)
- trim_right(field)

Function name: replace_chars

Replaces a range of characters in the required source field with a replacement character. Each group can consist of an optional replacement character, then a required start position and length. There are defaults for all arguments.

If the start position is not given, the value of one (1) is assumed. If the length is not given, the length of the original source string is used.

Source field:	${FIELDNAME}
[Replacement char]:	+
[Start and length]:	1,7

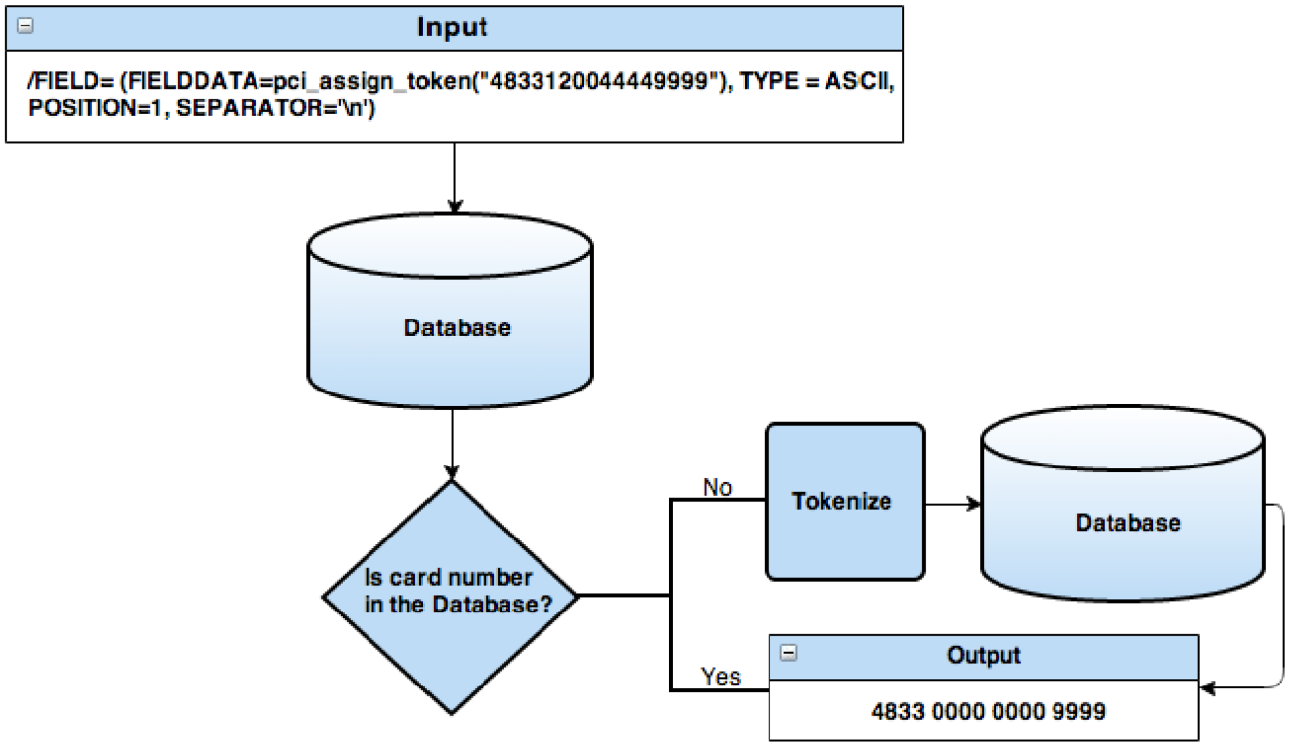
Type	Value
Number	1,7

```
/OUTFILE
    /ALI
    /PRO
    /FIE
    /FIE
    /FIE
    /FIE
    /FIE
    /FIE
    /FIE
    /FIE
    /FIE
    /FIE
    /FIE
    /FIE
    /FIE
    /FIE
    /FIE
    /FIELD=(ETHNIC_CODE, TYPE=ASCII, POSITION=16, SEPARATOR="|", EXT_FIELD="ETHNIC_CODE")
    /FIELD=(EMAIL_ADDRESS, TYPE=ASCII, POSITION=17, SEPARATOR="|", EXT_FIELD="EMAIL_ADDRESS")
    /FIELD=(MASK_ADDRESS_ID=replace_chars(ADDRESS_ID, "+"), TYPE=NUMERIC, POSITION=18, SEPARATOR="|", EXT_FIELD="ADDRESS_ID")
    /FIELD=(USER_STAMP, TYPE=ASCII, POSITION=19, SEPARATOR="|", EXT_FIELD="USER_STAMP")
    /FIELD=(DATE_STAMP, TYPE=ISO_DATE, POSITION=20, SEPARATOR="|", EXT_FIELD="DATE_STAMP")
    /FIELD=(TIME_STAMP, TYPE=NUMERIC, POSITION=21, SEPARATOR="|", EXT_FIELD="TIME_STAMP")
    /FIELD=(CREATED_BY, TYPE=ASCII, POSITION=22, SEPARATOR="|", EXT_FIELD="CREATED_BY")
    /FIELD=(CREATION_DATE, TYPE=ISO_DATE, POSITION=23, SEPARATOR="|", EXT_FIELD="CREATION_DATE")
    /FIELD=(UPDATED_BY, TYPE=ASCII, POSITION=24, SEPARATOR="|", EXT_FIELD="UPDATED_BY")
    /FIELD=(LAST_UPDATE_DATE, TYPE=ISO_DATE, POSITION=25, SEPARATOR="|", EXT_FIELD="LAST_UPDATE_DATE")
```

Tokenization

Tokenization is another type of data masking operation, and is sometimes specified by the payment card industry (PCI) to be used in lieu of, or in conjunction with, Primary Account Number (PAN) encryption.

Unlike encryption where data is protected, but is still present, a token acts as a link to where the data is stored. Authorized users can tokenize credit card values that may also be treated with a Format-Preserving Encryption (FPE) function first, and possibly de-tokenized later. The diagram below depicts the tokenization process.

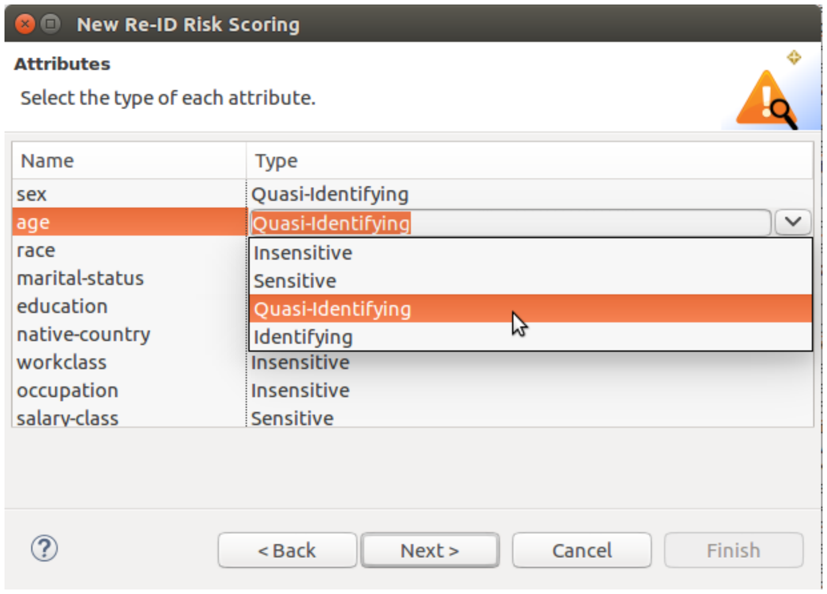


```
/FIELD= (FIELDDATA=pci_assign_token("4833120044449999"), TYPE = ASCII,
POSITION=1, SEPARATOR='\n')
```

Contact fieldshield@iri.com if you require another implementation of tokenization, or if you need help with or documentation on the use of FieldShield control language in searching selections, protecting, or reformatting data.

Re-ID Risk Scoring

To comply with the Health Insurance Portability and Accountability Act (HIPAA), you must de-identify both *key* identifiers and *quasi*-identifiers. Key identifiers are unique PHI values like name and Social Security Number, while quasi-identifiers are less unique attributes like age, race, state, gender, and occupation which can be used in tandem to identify unique records.



New Re-ID Risk Scoring

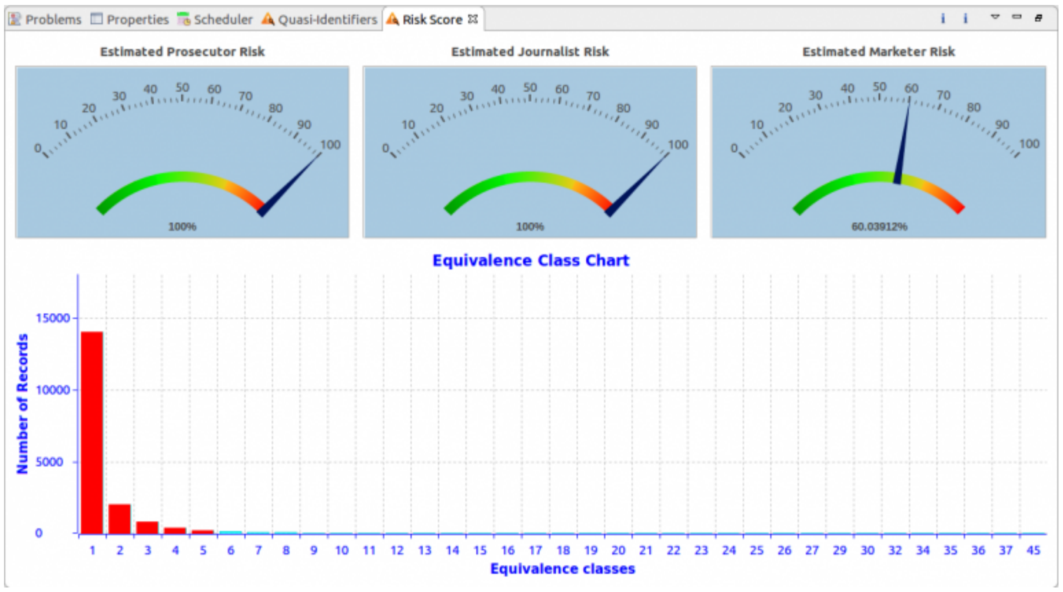
Attributes

Select the type of each attribute.

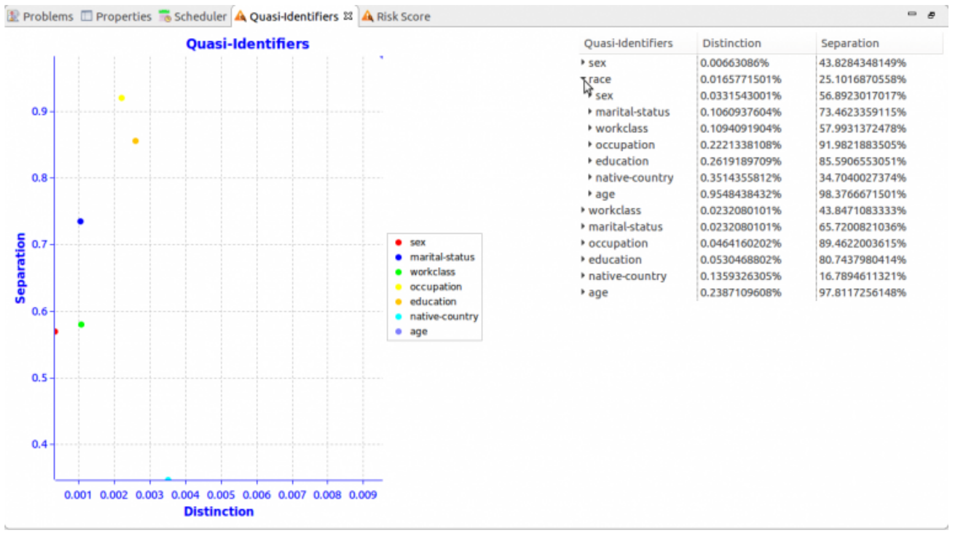
Name	Type
sex	Quasi-Identifying
age	Quasi-Identifying
race	
marital-status	
education	
native-country	
workclass	Insensitive
occupation	Insensitive
salary-class	Sensitive

Insensitive
Sensitive
Quasi-Identifying
Identifying

The Re-ID Risk Scoring wizard in IRI Workbench produces detailed and visual reports that statistically measure the risk of re-identification in three modes of attack and the number of records in each equivalence class:

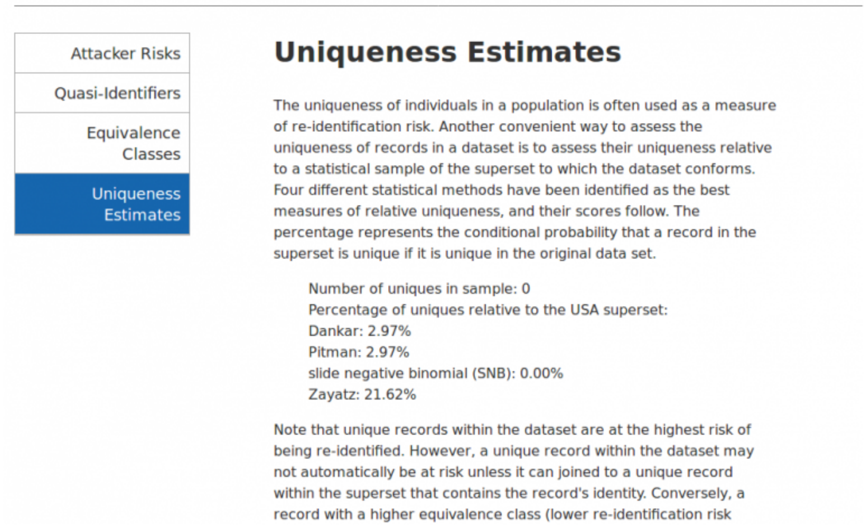


Another chart provides an interactive look at the different combinations of quasi-identifiers and their separation and distinction values to further assess their capacity to re-identify a record:



Quasi-Identifiers	Distinction	Separation
sex	0.00663086%	43.8284348149%
race	0.0165771501%	25.1016870558%
sex	0.0331543001%	56.8923017017%
marital-status	0.1060937604%	73.4623359115%
workclass	0.1094091904%	57.9931372478%
occupation	0.2221338108%	91.9821883505%
education	0.2619189709%	85.5906553051%
native-country	0.3514355812%	34.7040027374%
age	0.9548438432%	98.3766671501%
workclass	0.0232080101%	43.8471083333%
marital-status	0.0232080101%	65.7200821036%
occupation	0.0464160202%	89.4622003615%
education	0.0530468802%	80.7437980414%
native-country	0.1359326305%	16.7894611321%
age	0.2387109608%	97.8117256148%

In addition to those interactive graphs which can be saved in different image formats, the FieldShield re-ID risk determination report provides detailed descriptions of the metrics:



Uniqueness Estimates

The uniqueness of individuals in a population is often used as a measure of re-identification risk. Another convenient way to assess the uniqueness of records in a dataset is to assess their uniqueness relative to a statistical sample of the superset to which the dataset conforms. Four different statistical methods have been identified as the best measures of relative uniqueness, and their scores follow. The percentage represents the conditional probability that a record in the superset is unique if it is unique in the original data set.

Number of uniques in sample: 0
Percentage of uniques relative to the USA superset:
Dankar: 2.97%
Pitman: 2.97%
slide negative binomial (SNB): 0.00%
Zayatz: 21.62%

Note that unique records within the dataset are at the highest risk of being re-identified. However, a unique record within the dataset may not automatically be at risk unless it can joined to a unique record within the superset that contains the record's identity. Conversely, a record with a higher equivalence class (lower re-identification risk

After reviewing the risk scoring report in consultation with a qualified statistician whom IRI can also refer, you can create additional FieldShield jobs that generalize or blur one or more of the quasi-identifiers so they remain useful for research or marketing purposes, but less likely to lead to re-identification. See the section on generalization. After that, you can easily re-score re-generalized data sets from the attribute model you created in the first pass through the wizard.

For more information, see:
www.iri.com/blog/data-protection/hipaa-re-id-risk-scoring/

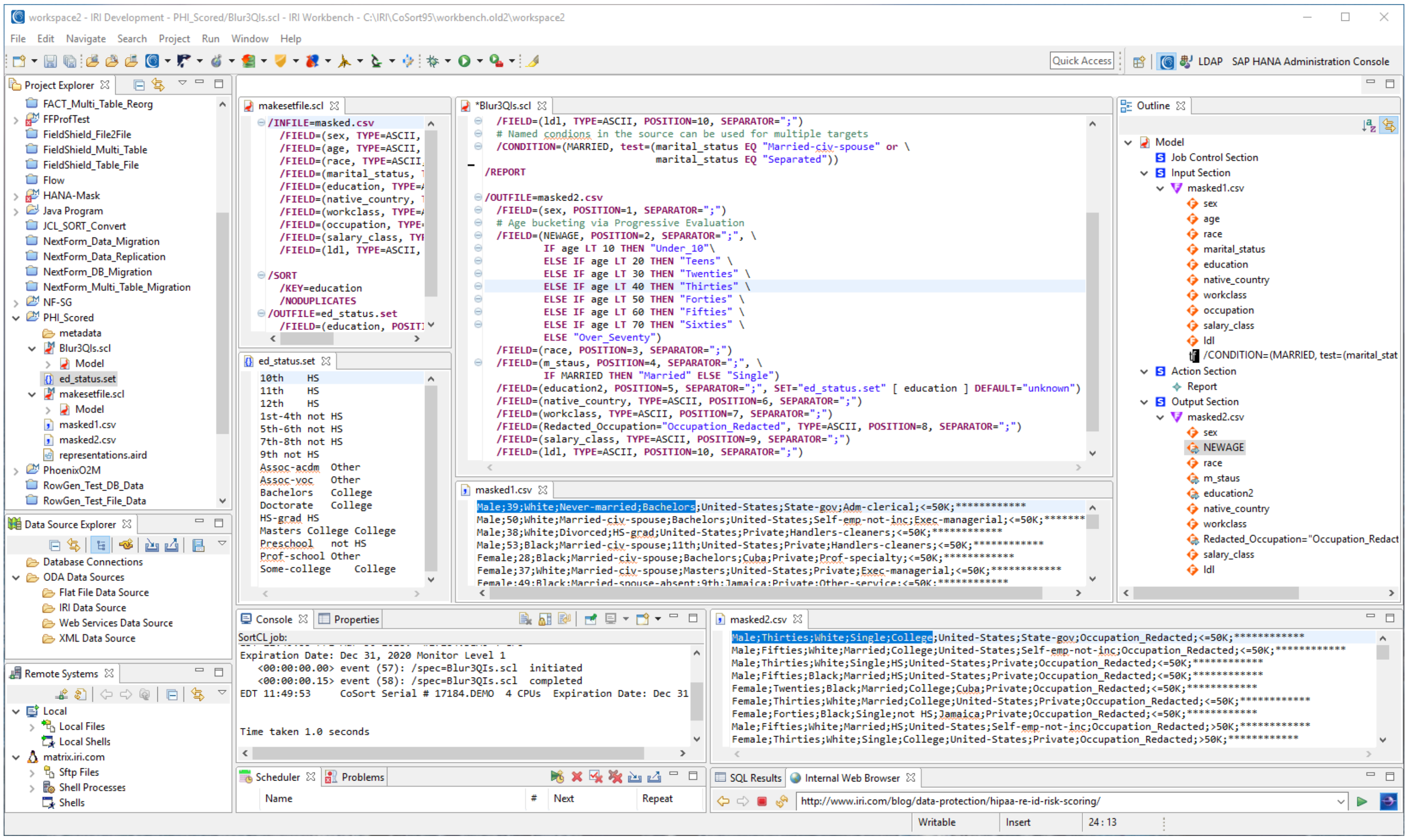
Anonymization (Blurring and Bucketing)

Quasi-identifying numeric values like age and date of birth, as well as descriptors like occupation and marital status, can all be used to re-identify people if there are enough of these attributes in the data set and/or they can be joined to a superset population with similar values.

For this reason, FieldShield jobs can apply one or more additional techniques to further anonymize or obfuscate the data, while still keeping it accurate enough for research or marketing purposes.

A numeric "smart blurring" function allows specific random values or values within defined ranges to be applied to ages and dates. This is also referred to as random noise.

In the example below however, specific ages are bucketed into decade groups, multiple marital status attributes are combined into two broader categories in a defined condition, educational attainments are simplified through a new set lookup file, and all occupations were explicitly redacted in place.


workspace2 - IRI Development - PHI_Scored/Blur3QIs.scl - IRI Workbench - C:\IRI\CoSort95\workbench.old2\workspace2
File Edit Navigate Search Project Run Window Help
Quick Access
LDAP SAP HANA Administration Console
Project Explorer
Data Source Explorer
Remote Systems
Outline
Console
Properties
Scheduler
Problems
SQL Results
Internal Web Browser

makesetfile.scl

```
/INFILE=masked.csv
  /FIELD=(sex, TYPE=ASCII,
  /FIELD=(age, TYPE=ASCII,
  /FIELD=(race, TYPE=ASCII,
  /FIELD=(marital_status,
  /FIELD=(education, TYPE=
  /FIELD=(native_country,
  /FIELD=(workclass, TYPE=
  /FIELD=(occupation, TYPE=
  /FIELD=(salary_class, TY
  /FIELD=(ldl, TYPE=ASCII,

/SORT
  /KEY=education
  /NODUPLICATES
/OUTFILE=ed_status.set
  /FIELD=(education, POSITI
```

ed_status.set

```
10th    HS
11th    HS
12th    HS
1st-4th not HS
5th-6th not HS
7th-8th not HS
9th not HS
Assoc-acdm  Other
Assoc-voc   Other
Bachelors   College
Doctorate   College
HS-grad HS
Masters College College
Preschool   not HS
Prof-school Other
Some-college    College
```

*Blur3QIs.scl

```
  /FIELD=(ldl, TYPE=ASCII, POSITION=10, SEPARATOR=";")
  # Named condions in the source can be used for multiple targets
  /CONDITION=(MARRIED, test=(marital_status EQ "Married-civ-spouse" or \
                             marital_status EQ "Separated"))
/REPORT

/OUTFILE=masked2.csv
  /FIELD=(sex, POSITION=1, SEPARATOR=";")
  # Age bucketing via Progressive Evaluation
  /FIELD=(NEWAGE, POSITION=2, SEPARATOR=";", \
          IF age LT 10 THEN "Under_10"\
          ELSE IF age LT 20 THEN "Teens" \
          ELSE IF age LT 30 THEN "Twenties" \
          ELSE IF age LT 40 THEN "Thirties" \
          ELSE IF age LT 50 THEN "Forties" \
          ELSE IF age LT 60 THEN "Fifties" \
          ELSE IF age LT 70 THEN "Sixties" \
          ELSE "Over_Seventy")
  /FIELD=(race, POSITION=3, SEPARATOR=";")
  /FIELD=(m_staus, POSITION=4, SEPARATOR=";", \
          IF MARRIED THEN "Married" ELSE "Single")
  /FIELD=(education2, POSITION=5, SEPARATOR=";", SET="ed_status.set" [ education ] DEFAULT="unknown")
  /FIELD=(native_country, TYPE=ASCII, POSITION=6, SEPARATOR=";")
  /FIELD=(workclass, TYPE=ASCII, POSITION=7, SEPARATOR=";")
  /FIELD=(Redacted_Occupation="Occupation_Redacted", TYPE=ASCII, POSITION=8, SEPARATOR=";")
  /FIELD=(salary_class, TYPE=ASCII, POSITION=9, SEPARATOR=";")
  /FIELD=(ldl, TYPE=ASCII, POSITION=10, SEPARATOR=";")
```

masked1.csv

```
Male;39;White;Never-married;Bachelors;United-States;State-gov;Adm-clerical;<=50K;************
Male;50;White;Married-civ-spouse;Bachelors;United-States;Self-emp-not-inc;Exec-managerial;<=50K;*******
Male;38;White;Divorced;HS-grad;United-States;Private;Handlers-cleaners;<=50K;************
Male;53;Black;Married-civ-spouse;11th;United-States;Private;Handlers-cleaners;<=50K;************
Female;28;Black;Married-civ-spouse;Bachelors;Cuba;Private;Prof-specialty;<=50K;************
Female;37;White;Married-civ-spouse;Masters;United-States;Private;Exec-managerial;<=50K;************
Female;49;Black;Married-spouse-absent;9th;Jamaica;Private;Other-service;<=50K;************
```

Outline

```
Model
  Job Control Section
  Input Section
    masked1.csv
      sex
      age
      race
      marital_status
      education
      native_country
      workclass
      occupation
      salary_class
      ldl
      /CONDITION=(MARRIED, test=(marital_stat
  Action Section
    Report
  Output Section
    masked2.csv
      sex
      NEWAGE
      race
      m_staus
      education2
      native_country
      workclass
      Redacted_Occupation="Occupation_Redact
      salary_class
      ldl
```

Console — SortCL job:

```
Expiration Date: Dec 31, 2020 Monitor Level 1
   <00:00:00.00> event (57): /spec=Blur3QIs.scl  initiated
   <00:00:00.15> event (58): /spec=Blur3QIs.scl  completed
EDT 11:49:53     CoSort Serial # 17184.DEMO  4 CPUs  Expiration Date: Dec 31


Time taken 1.0 seconds
```

masked2.csv

```
Male;Thirties;White;Single;College;United-States;State-gov;Occupation_Redacted;<=50K;************
Male;Fifties;White;Married;College;United-States;Self-emp-not-inc;Occupation_Redacted;<=50K;************
Male;Thirties;White;Single;HS;United-States;Private;Occupation_Redacted;<=50K;************
Male;Fifties;Black;Married;HS;United-States;Private;Occupation_Redacted;<=50K;************
Female;Twenties;Black;Married;College;Cuba;Private;Occupation_Redacted;<=50K;************
Female;Thirties;White;Married;College;United-States;Private;Occupation_Redacted;<=50K;************
Female;Forties;Black;Single;not HS;Jamaica;Private;Occupation_Redacted;<=50K;************
Male;Fifties;White;Married;HS;United-States;Self-emp-not-inc;Occupation_Redacted;>50K;************
Female;Thirties;White;Single;College;United-States;Private;Occupation_Redacted;>50K;************
```

http://www.iri.com/blog/data-protection/hipaa-re-id-risk-scoring/

The new result set can now be re-run through the risk scoring wizard to produce another determination of re-identification risk based on now less distinct quasi-identifying attributes.

Software Development Kit / DDM

The software development kit (SDK) for dynamic data masking (DDM) or unmasking within applications uses APIs written in Java and .NET.

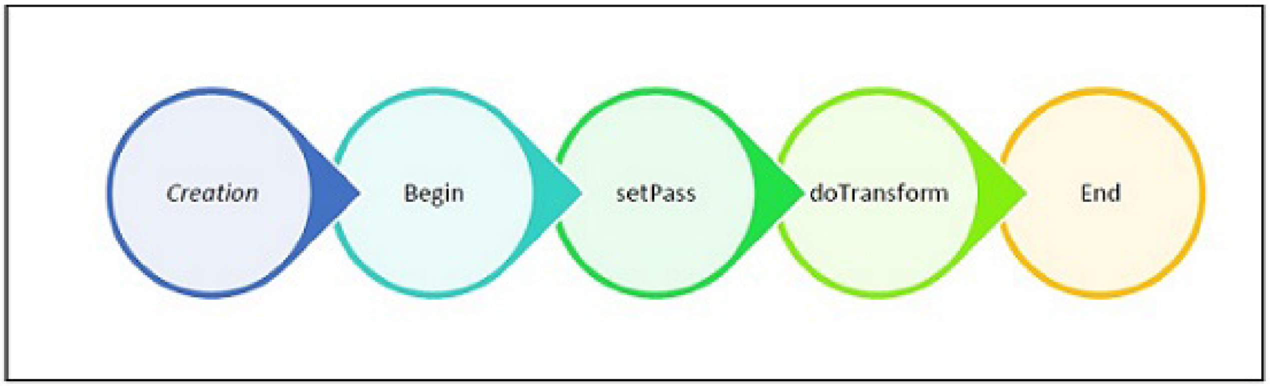
The FieldShield library is a group of classes that provides several ways of protecting or altering data by using encryption/decryption processes, hashing, masking, and base64 to hex conversion. These functions are compatible with those used in the standalone, static data masking executable. It is therefore possible to encrypt data at rest in a database in a standalone FieldShield operation, and selectively mask it in transit from an application calling a corresponding decryption function in the SDK.

FieldShield SDK encryption and decryption algorithms include:

- AES-256 (Advanced Encryption Standard); the state-of-the-art and most powerful of the encryption algorithms.
- FP-ASCII (Format Preserved for American Standard Code for Information Interchange); the original format data is preserved but still protected with AES-256 bit encryption.
- FP- ALPHANUM; another version of the Format Preserving algorithm above, in which it can be set to number mode only or alphanumeric for data preservation.

FieldShield Library Process

Each function in the FieldShield library flows in a defined manner. The figure below shows the sequence of methods for each encryption and decryption usage. Once the password is set for a given object, it does not need to be set each time the encryption object is used.



The other classes follow a similar flow in usage, varying slightly. The steps are defined thus:

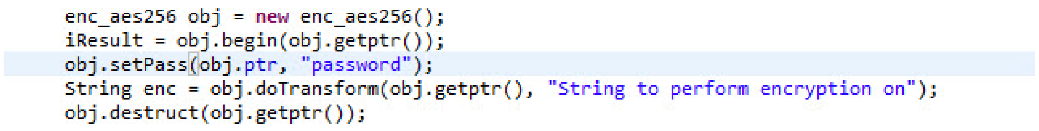
1. **Creation -** Create an instance of the desired class. Most languages use the keyword new.
2. **Begin -** After creating an instance from the class, use this instance to call the begin() method. This function will initiate an encryption or decryption process.

3. **SetPass -** Next, the pass phase must be set by calling the setPass method. This method requires a String to be passed in as a parameter.
4. **DoTransform -** Once the passphrase has been set, the core method, doTransform(Data), is called to encrypt or decrypt. A String is passed as a parameter, which is then encrypted or decrypted using the algorithm chosen. For the alphanumeric class, a numeric doTransform method exists to be used on numerals. The method will return the data that has been encrypted or decrypted with the appropriate type.
5. **Destruct -** Finally, call the method destruct() to release the object from the occupied space in memory. Since it uses the DLL file, it is necessary to destruct the object.

6.

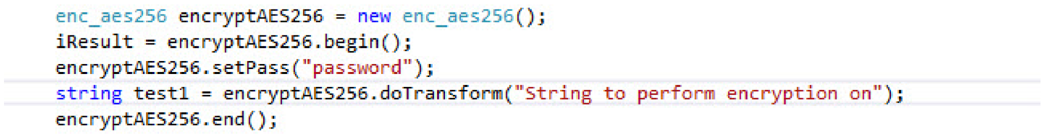
Code Samples

Java Example



```
enc_aes256 obj = new enc_aes256();
iResult = obj.begin(obj.getptr());
obj.setPass(obj.ptr, "password");
String enc = obj.doTransform(obj.getptr(), "String to perform encryption on");
obj.destruct(obj.getptr());
```

.NET Example in C#



```
enc_aes256 encryptAES256 = new enc_aes256();
iResult = encryptAES256.begin();
encryptAES256.setPass("password");
string test1 = encryptAES256.doTransform("String to perform encryption on");
encryptAES256.end();
```

Technical Specifications

FieldShield is a program for protecting Personally-Identifying Information (PII), Protected Health Information (PHI), and other sensitive data using highly efficient, targeted data masking functions according to business rules. It can nullify the effect of data breaches by protecting PII at the field level across multiple data sources and facilitate compliance with U.S. and international data privacy laws.

FieldShield can also provide file-level protection with one encryption algorithm and a key. However, most FieldShield users apply multiple functions to specific elements; i.e., columns in database tables and fields in files that are intended for specific uses and recipients.

To safeguard sensitive data, FieldShield offers four significant features that enable you to do the following in any combination:

- Choose one or more protection (e.g., encryption) functions in combination
- Apply theses functions at the field (column) or record (row) level
- Make secure data look real
- Generate secure output targets in one pass through the input data

Installation

- Distributes via Internet or user-specified media
- Loads in under two minutes
- Uses menu-driven setup and configuration utility

Invocation

- Command line (including pipe sequences), shell commands and batch scripts
- IRI Workbench – Eclipse GUI for Windows (JRE required)
- Application calling programs as a standalone executable, subroutine or coroutine call, with or without additional exit routines

Ease of Use

- Processes data using record layouts and SQL-like field descriptions within applications or centralized data definition files (DDF)
- Provides on-line help, pre-runtime application validation, and runtime errors
- Leverages centralized application and file layout definitions (metadata repositories)
- Reports problems to standard error when invoked from a program, or to an error log
- Runs silently or with verbose messaging without user intervention
- Allows user control over the amount of informational output produced
- Generates a query-ready XML audit log for data forensics and privacy compliance
- Describes commands and options through on-line documentation
- Fully supports job design, execution, and modification in familiar Eclipse GUI
- Easy-to-use interfaces preclude the need for training classes (advanced training is available in Florida or at user sites)
- Phone, web, and email support available directly from the product developers
- Local language support is available from more than 30 international offices

Functions

- Data discovery and classification through multiple search and profiling wizards
- Multiple encryption via AES-256 routines (e.g., FPE_ASCII / FPE_ALPHANUM) or a user-supplied routine
- GPG/PGP encryption and decryption with GPG key ring management support
- ASCII de-identification, encoding, hashing, and randomization
- Reversible and non-reversible pseudonymization
- Redaction (partial or full string masking) or conditional value removal
- Anonymization via generalization, blurring, or custom obfuscation
- Re-ID risk determination and reporting through a fit-for-purpose scoring wizard
- Audit logging to verify compliance, and statistical reporting for performance review

Data Sources

Sources

Local pipes via stdin, plus any number of structured (COBOL, CSV, LDIF, fixed, etc.), and *some* semi-structured (flat JSON and XML, Excel, ASN.1 CDR) files on-premise *or* in Amazon S3, Azure Blob or GCP storage buckets. Also supported are named pipes (on UNIX or Linux), and ODBC-connected relational databases on premise or in the cloud. HDFS, MongoDB, HIVE, MQTT, Kafka, HTTP/S and FTP/S are also accessible through URL specifications.

Input Size

No limit

Record/Row Length

1 ≤ fixed-length ≤ 65,535 bytes
0 ≤ variable-length ≤ 65,535 bytes

Program

Optional, user-written routines for reading special sources

Data Types

The following categories of data types are supported:

Character strings	ASCII, EBCDIC, ASCII in EBCDIC order, days of the week, months of the year, etc.
Numerics	ASCII-Numeric, MF COBOL types, RM COBOL types, integers, floats, doubles, zoned decimal, etc
Timestamps	American, European, ISO, Japanese, and current time, date, and timestamp.
Multi-byte	Double-byte types such as Shift JIS, EHANGUL, and Big 5.

Targets

Terminal
stdout

Table
ODBC- and JDBC-connected databases on premise or in the cloud

File
User-selected name(s) or device(s), including input file overwrite and create or append logic, plus named and unnamed pipes. Target files can persist in the LAN or cloud storage buckets (specifically, in Amazon S3, Azure Blob, and Google Cloud Platform)..

Both
Standard output and file to observe and abort unintended executions

URLs
HDFS, MongoDB, HIVE, MQTT, Kafka, and FTP/S

Report
Customized detail report with header, footer, page count, labeling, and other formatting features

Program
Optional, user-written routines for writing to special targets

Operating Optional Modes
- standalone or embedded batch execution
- user program integration

Platforms

- AIX 6.1 and above on IBM Power architecture
- HP-UX 11.31b on ia64 architecture
- Linux kernel 2.6 and above on Intel x86 & x64 architecture
- Linux kernel 3.6 and above on ARMv6 and above architecture
- macOS
- Solaris 5.8 and above on Sun SPARC architecture
- Solaris 10 on Intel x64 architecture
- Windows XP and above on Intel x86 & x64 architecture
- Windows Server 2003 and above on Intel x86 & x64 architecture

Licensing Information

IRI and its representatives around the world license FieldShield for leased or perpetual use on individual computer systems. Maintenance (technical support and site-specific software updates) services are provided free of charge during the first year after installation. Subsequent annual maintenance is usually offered at 20% of the base license fee, and 24/7 technical support is available.

Discounts are available for multiple copies of FieldShield at the same site, distributed anonymization, and for runtime integration and redistribution (ISVs only). IRI is generous with credit for hardware upgrades and migrations, and provides for no- or low-cost failover (disaster recovery) licenses.

U.S. educational and 501c(3) non-profit institutions qualify for a 10% license fee discount, and government agencies can buy FieldShield from prime contractors on GSA schedule.

FieldShield is also a product member of the IRI Data Protector Suite and an included component in the IRI Voracity total data management platform (subscription). Therefore, its functionality can be licensed in a discounted bundle with other IRI data-centric security software.

Professional Services

An IRI professional services engagement allows you to leverage more than 100 collective years of IT and integrated data-handling experience. Examples of IRI professional services engagements involve:

- *Big data preparation* – packaging, protecting, and provisioning structured and unstructured data sets for analytic/BI, database (DB), and ETL operations
- *Data masking* – profiling, de-identification, encryption, tokenization and other services to aid data loss prevention, data governance, and data privacy law compliance efforts
- *Risk scoring and certification* -- partner statistical analysis and legal advice pertaining to data re-identification, HIPAA compliance, insurance coverage, and breach defense
- *Data replication and federation* – acquiring, re-mapping, and creating views
- *DB migration* – mapping table data and relations to new versions/platforms
- *Data conversion* – reformatting LDIF, XML, and COBOL index files (e.g., Vision, MF-ISAM), multi-byte character sets, most mainframe data types, and endian states
- *Master data management* – value discovery, format definition, validation and enrichment, security, and lineage
- *Program replacement* – translating cryptic and inefficient SQL, 3GL, ETL, legacy sort, and shell procedures into IRI's simple, portable, 4GL text scripts
- *Test data management* – end-to-end services from DevOps needs definition through data generation and target persistence (resembling production data)

IRI FieldShield
PII / PHI Classification & Masking

INNOVATIVE ROUTINES INTERNATIONAL (IRI), INC.

Suite 303, Atlantis Center
2194 Highway A1A
Melbourne, FL 323937-4932 USA
Phone | +1.321.777.8889
www.iri.com/fieldshield

IRI
Total Data Management