Article

Henry Pereira · Dec 1, 2021 5m read

Open Exchange

Data anonymization, introducing iris-Disguise

First of all, what is data anonymization?

According to Wikipedia:

> **Data anonymization** is a type of information sanitization whose intent is privacy protection. It is the process of removing personally identifiable information from data sets, so that the people whom the data describe remain anonymous.

In other words, the data anonymization is a process that retains the data but keeps the source anonymous. Depending on the adopted anonymization technique the data is redacted, masked or substituted.

And that is the purpose of iris-Disguise, to provide a set of anonymization tools.

You can use in two different ways, by method execution or specify your anonymization strategy inside the persistent class definition itself.

The current version of iris-Disguise offers 6 strategies to anonymize data:

- **Destruction**
- **Scramble**
- **Shuffling**
- **Partial Masking**
- **Randomization**
- **Faking**

Let me explain each strategy, I will show a method execution with an example and as mentioned, I'll also show how to apply inside the persistent class definition.
To use **iris-Disguise** in this way you need to "*wear a disguise glasses*".
In the persistent class, you can extent the **dc.Disguise.Glasses** class and change any property with the data type with the strategy of your choice.
After that, at any moment, just call the **DisguiseProcess** method on the class. All the values will be replaced using the strategy of the data type.

So buckle up and let's go.

Destruction

This strategy will replace a entire column with a word ('CONFIDENTIAL' is the default).

```
Do ##class(dc.Disguise.Strategy).Destruction("classname", "propertyname", "Word to re
place")
```

The third parameter is optional. If not provided, the word 'CONFIDENTIAL' will be used.

```
Class packageSample.FictionalCharacter Extends (%Persistent, dc.Disguise.Glasses)
{
Property Name As dc.Disguise.DataTypes.String(FieldStrategy = "DESTRUCTION");
}
```

```
Do ##class(packageSample.FictionalCharacter).DisguiseProcess()
```

ID	Name
1	Leonardo
2	Donatello
3	Michelangelo
4	Raphael

ID	Name
1	CONFIDENTIAL
2	CONFIDENTIAL
3	CONFIDENTIAL
4	CONFIDENTIAL

Scramble

This strategy will scrambling all characters in a property.

```
Do ##class(dc.Disguise.Strategy).Scramble("classname", "propertyname")
```

```
Class packageSample.FictionalCharacter Extends (%Persistent, dc.Disguise.Glasses)
{
Property Name As dc.Disguise.DataTypes.String(FieldStrategy = "SCRAMBLE");
}
```

```
Do ##class(packageSample.FictionalCharacter).DisguiseProcess()
```

ID	Name
1	Leonardo
2	Donatello
3	Michelangelo
4	Raphael

ID	Name
1	enodaroL
2	ooelantID
3	lieanMclehog
4	aplehaR

Shuffling

Shuffling will rearrange all values in a given property. Is not a masking strategy because it works "verticaly".
This strategy is useful for relatinship because referential integrity will be kept.
Until this version, this method only works on **one-to-many relationships**.

```
Do ##class(dc.Disguise.Strategy).Shuffling("classname", "propertyname")
```

```
Class packageSample.FictionalCharacter Extends (%Persistent, dc.Disguise.Glasses)
{
Property Name As %String;
Property Weapon As dc.Disguise.DataTypes.String(FieldStrategy = "SHUFFLING");
}
```

```
Do ##class(packageSample.FictionalCharacter).DisguiseProcess()
```

ID	Name	Weapon
1	Leonardo	Katana
2	Donatello	Bo
3	Michelangelo	Nunchaku
4	Raphael	Sai

ID	Name	Weapon
1	Leonardo	Sai
2	Donatello	Nunchaku
3	Michelangelo	Katana
4	Raphael	Bo

Partial Masking

This strategy will obfuscate the part of data, a credit card number for example, can be replaced by 456X XXXX XXXX X783

```
Do ##class(dc.Disguise.Strategy).PartialMasking("classname", "propertyname", prefixLength, suffixLength, "mask")
```

PrefixLength, suffixLength and mask are optional. If not provided, the default values will be used.

```
Class packageSample.FictionalCharacter Extends (%Persistent, dc.Disguise.Glasses)
{
Property Name As %String;
Property SSN As dc.Disguise.DataTypes.PartialMaskString(prefixLength = 2, suffixLength = 2);
Property Weapon As %String;
}
```

```
Do ##class(packageSample.FictionalCharacter).DisguiseProcess()
```

ID	Name	SSN	Weapon
1	Leonardo	440-15-5966	Katana
2	Donatello	574-15-7023	Bo
3	Michelangelo	417-29-9142	Nunchaku
4	Raphael	035-01-5028	Sai

ID	Name	SSN	Weapon
1	Leonardo	44X-XX-XX66	Katana
2	Donatello	57X-XX-XX23	Bo
3	Michelangelo	41X-XX-XX42	Nunchaku
4	Raphael	03X-XX-XX28	Sai

Randomization

This strategy will generate purely random data. There are three types of randomization: integer, numeric and date.

```
Do ##class(dc.Disguise.Strategy).Randomization("classname", "propertyname", "type", from, to)
```

type: "integer", "numeric" or "date". "integer" is the default.

from and to are optional. Is to define the range of randomization.
For integer type the default range is 1 to 100. For numeric type the default range is 1.00 to 100.00.

```
Class packageSample.FictionalCharacter Extends (%Persistent, dc.Disguise.Glasses)
{
Property Name As %String;
Property Age As dc.Disguise.DataTypes.RandomInteger(MINVAL = 10, MAXVAL = 25);
Property SSN As %String;
Property Weapon As %String;
}
```

```
Do ##class(packageSample.FictionalCharacter).DisguiseProcess()
```

ID	Name	Weapon	Age
1	Leonardo	Katana	14
2	Donatello	Bo	14
3	Michelangelo	Nunchaku	13
4	Raphael	Sai	15

ID	Name	Weapon	Age
1	Leonardo	Katana	22
2	Donatello	Bo	11
3	Michelangelo	Nunchaku	17
4	Raphael	Sai	19

Fake Data

The idea of Faking is to replace data with random but plausible values.
iris-Disguise provides a small set of methods to generate fake data.

```
Do ##class(dc.Disguise.Strategy).Fake("classname", "propertyname", "type")
```

type: "firstname", "lastname", "fullname", "company", "country", "city" and "email"

```
Class packageSample.FictionalCharacter Extends (%Persistent, dc.Disguise.Glasses)
{
Property Name As dc.Disguise.DataTypes.FakeString(FieldStrategy = "FIRSTNAME");
Property Age As %Integer;
Property SSN As %String;
Property Weapon As %String;
}
```

```
Do ##class(packageSample.FictionalCharacter).DisguiseProcess()
```

ID	Name	Weapon	SSN	Age
1	Leonardo	Katana	440-15-5966	14
2	Donatello	Bo	574-15-7023	14
3	Michelangelo	Nunchaku	417-29-9142	13
4	Raphael	Sai	035-01-5028	15

ID	Name	Weapon	SSN	Age
1	Quigley	Katana	440-15-5966	14
2	David	Bo	574-15-7023	14
3	Francis	Nunchaku	417-29-9142	13
4	Tara	Sai	035-01-5028	15

I want to hear from you!

Feedback and ideas are welcome!

Let me know what you think of this tool, how it fits your needs and what features are missing.

And I want to say a very special thanks to @Henrique Dias, @Oliver Wilms, @Robert Cemper, @Yuri Marx and @Evgeny Shvarov that commented, reviewed, suggested and made rich discussions which inspired me to create and improve the iris-Disguise.

#InterSystems IRIS
Check the related application on InterSystems Open Exchange

Source URL: https://community.intersystems.com/post/data-anonymization-introducing-iris-disguise