

記事

[Hiroshi Sato](#) · 2021年1月27日 4m read

クラス定義のプロパティ表示順と、プロパティに対応したグローバル変数の格納順

これは [InterSystems FAQ サイト](#)の記事です。

クラス定義のプロパティの表示順は、スタジオのプロパティウィザードを利用して登録した場合は、末尾に追記されます。

また、エディタ上の任意の場所でプロパティ定義文を記述する場合は、その場所に追記され、クラス定義が登録されます。

つまり、定義者が記述した順番に登録されます。

(スタジオが並び換えを行ったりはしません。)

作成したクラス定義が、PersistentやSerialのようにデータベースに格納する属性を持ったクラス定義である場合、“初回のコンパイル”でクラス定義に対応するグローバル変数の定義情報 = ストレージ定義を作成します。

初回コンパイル以降に、プロパティ定義の追加が行われれば、そのプロパティに対応するグローバル変数のスロット番号を、末尾に追加し、ストレージ定義を更新します。

以下の例は、クラス定義に対応するストレージ定義の例です。

(初回コンパイル時の状態)

```
Class Sample.Person Extends %Persistent
{ Property Name As %String; /// ???
Property DOB As %Date; <storage name="Default">
<data name="PersonDefaultData">
<value name="1">
<value>%CLASSNAME</value></value> <value name="2">
<value>Name</value></value> <value name="3">
<value>DOB</value></value></data> <datalocation>^Sample.PersonD</datalocation>
<defaultdata>PersonDefaultData</defaultdata>
<extentsize>100000</extentsize>
<idlocation>^Sample.PersonD</idlocation>
<indexlocation>^Sample.PersonI</indexlocation>
<streamlocation>^Sample.PersonS</streamlocation>
<type>%Library.CacheStorage</type></storage> }
```

<storage name="Default">から</storage>までの表示が、クラス定義の初回コンパイルで作成されるストレージ定義情報です。

作成したSample.Personクラスの格納先グローバル変数は、ストレージ定義の<datalocation> <indexlocation> <streamlocation> を参照するとわかります。

また、各プロパティ定義が、指定グローバル変数のどこに格納されるかは、<value name="2"> と <value name="3">を参照するとわかります。

```
<value name="2">
<value>Name</value>&lt;/value> <value name="3">
<value>DOB</value>&lt;/value>
```

つまり、Nameプロパティは、^Sample.PersonDの \$ListBuild()構造の2番目に格納され、DOBは3番目に格納されることがわかります。

ここで、Addressプロパティを、クラス定義の表示上、一番上に追加します。

```
Class Sample.Person Extends %Persistent
{ Property Address As %String; Property Name As %String; /// ???
Property DOB As %Date; <storage name="Default">
<data name="PersonDefaultData">
<value name="1">
<value>%%CLASSNAME</value>&lt;/value> <value name="2">
<value>Name</value>&lt;/value> <value name="3">
<value>DOB</value>&lt;/value> <value name="4">
<value>Address</value>&lt;/value>&lt;/data> <datalocation>^Sample.PersonD</datalocation>
<defaultdata>PersonDefaultData</defaultdata>
<extentsize>100000</extentsize>
<idlocation>^Sample.PersonD</idlocation>
<indexlocation>^Sample.PersonI</indexlocation>
<streamlocation>^Sample.PersonS</streamlocation>
<type>%Library.CacheStorage</type> }
```

ストレージ定義情報を参照すると、クラス定義の表示上1番上に登録したAddressプロパティは、

```
<value name="4">
<value>Address</value>&lt;/value>
```

\$ListBuildの4番目に格納される定義として追加されています。

つまり、ストレージ定義は、クラス定義の表示上、どこに追記されても、後から追加されたプロパティについては、格納位置として、一番最後に追加していく仕組みがわかります。

ということで、クラス定義上の表示順と、ストレージ定義の格納順は必ずしも一致しない事があります。

なお、ストレージの格納順序を変更することはできます。

以下の例は、AddressとNameの格納順を、

現在

Address 4番目

Name 2番目

変更後

Address 2番目

Name 4番目

に変更した状態のストレージ定義例です。(一部抜粋)

```
<data name="PersonDefaultData">
<value name="1">
<value>%CLASSNAME</value></value> <value name="4">
<value>Name</value></value> <value name="3">
<value>DOB</value></value> <value name="2">
<value>Address</value></value></data>
```

ご覧いただいた通り、AddressプロパティとNameプロパティのスロット番号を入れ替えるだけで、ストレージ定義の格納順の変更が行えます。

ただ、ストレージ定義情報の順番のみが入れ替わるだけであり、既存グローバル変数のデータが入れ替わる事はありません。

この点、ご注意ください。

(修正したストレージ定義に合わせて、グローバル変数を入れ替えたい場合は、手動でグローバル変数の中身を変更する必要があります。)

[#スタジオ](#) [#Caché](#) [#Ensemble](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

ソースURL:

<https://jp.community.intersystems.com/post/%E3%82%AF%E3%83%A9%E3%82%B9%E5%AE%9A%E7%BE%A9%E3%81%AE%E3%83%97%E3%83%AD%E3%83%91%E3%83%86%E3%82%A3%E8%A1%A8%E7%A4%BA%E9%A0%86%E3%81%A8%E3%80%81%E3%83%97%E3%83%AD%E3%83%91%E3%83%86%E3%82%A3%E3%81%AB%E5%AF%BE%E5%BF%9C%E3%81%97%E3%81%9F%E3%82%B0%E3%83%AD%E3%83%BC%E3%83%90%E3%83%AB%E5%A4%89%E6%95%B0%E3%81%AE%E6%A0%BC%E7%B4%8D%E9%A0%86>