

Article

[Lorenzo Scalese](#) · Avr 8, 2022 2m de lecture

Persistence des propriétés multidimensionnelles - Partie 2 (Nouvel Ère)

Alors que la solution classique suivait d'assez près les concepts et le design de ses ancêtres, Caché / IRIS permet une approche plus moderne des propriétés flexibles/multidimensionnelles.

Voyons notre classe de démonstration comme avant :

```
Class DC.Multi Extends (%Persistent, %Populate) [ Final ]
{
Property Name As %String;
Property DOB As %Date;
Property mJSON As %DynamicObject;
/// Property Multi As %String [ MultiDimensional ];
```

L'objet JSON nous permet toute la flexibilité dont on peut avoir besoin avec le paradigme clé-valeur comme concept de base. La carte de stockage reflète cette situation par une entrée appropriée.

Storage Default

```
{
<Data name="MultiDefaultData">
  <Value name="1">
    <Value>Name</Value>
  </Value>
  <Value name="2">
    <Value>DOB</Value>
  </Value>
</Data>
<Data name="mJSON">
  <Attribute>mJSON</Attribute>
  <Structure>node</Structure>
  <Subscript>"mJSON"</Subscript>
</Data>
```

```
<DataLocation>^DC.MultiD</DataLocation>  
<DefaultData>MultiDefaultData</DefaultData>  
<IdLocation>^DC.MultiD</IdLocation>  
<IndexLocation>^DC.Multil</IndexLocation>  
<StreamLocation>^DC.MultiS</StreamLocation>  
<Type>%Storage.Persistent</Type>  
}
```

Diagnostic : Le générateur de stockage a déjà prévu la structure que nous devons auparavant ajouter manuellement.

Problème n°1) n'existe plus

La même chose pour le n°2) pas de bidouillage supplémentaire pour SQL.
L'accès est prêt à l'emploi.

ID	DOB	Name	mJSON
1	08/07/1981	Braam,Ted Q.	{}
2	01/03/2012	Klingman,Uma C.	{"A":"ahahah","Rob":"VIP","Rob2":1111,"Rob3":true}
3	06/25/1966	Goldman,Kenny H.	{}

Je reconnais que ma fantaisie pour les données de test était plutôt limitée. 😊

Quel est le prix de cette amélioration ?

Au lieu de : `Set obj.Multi("robert")="rcc"`

c'est maintenant : `Do obj.mJSON.%Set("robert","rcc")`

une sous-structure comparable à `obj.Multi("robert",1)` pourrait nécessiter une liste dynamique

au lieu d'une navigation par `$order()` et `$query()` vous utilisez maintenant un itérateur et opérez de manière totalement compatible avec JSON.

récupérer des données par `set var=obj.Multi("robert")`

ou des changements similaires à `set var=obj.mJSON.%Get("robert")`

Personnellement, je trouve cette approche beaucoup plus claire et indépendante des objets enregistrés ou persistants.

[#JSON](#) [#ObjectScript](#) [#SQL](#) [#Valeur clé](#) [#Caché](#) [#Ensemble](#) [#InterSystems](#) [IRIS](#)

URL de la
source: <https://fr.community.intersystems.com/post/persistence-des-propri%C3%A9t%C3%A9s-multidimensionnelles-partie-2-nouvel-%C3%A8re>