

El arte de mapear Globals para Clases (3 de 3)

Artículo

[Estevan Martinez](#) · Jul 1, 2019



Lectura de 7 min

El arte de mapear Globals para Clases (3 de 3)

¿Alguno de ustedes desea traer de vuelta una antigua aplicación en MUMPS? Siga estos pasos para mapear sus globales ya establecidos para clases y mostrar todos esos hermosos datos en Objects y SQL.

Si lo anterior no le suena conocido, comience leyendo estos artículos previos:

- [El Arte del Mapeo de Globales para Clases \(1 de 3\)](#)
- [El Arte del Mapeo de Globales para Clases \(2 de 3\)](#)

En este ejemplo mostraré cómo mapear una estructura clásica de padre e hijo.

La misma aclaración: Si sus variables globales todavía no tienen ni pies ni cabeza después de revisar estos artículos, póngase en contacto con el WRC y trataremos de ayudarlo a resolver su problema:

Support@InterSystems.com.

Pasos para mapear una variable global para una clase:

1. Identifique algún patrón que se repita en los datos globales
2. Identifique qué la hace una clave única
3. Identifique las propiedades y sus categorías
4. Defina las propiedades en la clase (no olvide las propiedades de los subíndices de las variables)
5. Defina el índice IdKey
6. Establezca la definición de almacenamiento:
 1. Establezca los subíndices, e incluya la IdKey
 2. Establezca una sección para los datos
 3. Ignore la sección "Row ID (Identificación de la fila)". El 99% de las veces el valor que desea es el predeterminado, así que deje que el sistema lo llene por usted
7. Compile y pruebe su clase / tabla

En el ejemplo anterior cada persona tenía 1 actividad, pero eso era bastante aburrido, así que ahora queremos que la gente tenga más de una actividad. En el siguiente ejemplo, el paso 1 se vuelve un poco más complejo porque hay 2 conjuntos diferentes de datos repetidos.

```
^ParentChild(1)="Brendan^45956"  
^ParentChild(1,"Hobbies",1)="Pit Crew"  
^ParentChild(1,"Hobbies",2)="Kayaking"  
^ParentChild(1,"Hobbies",3)="Skiing"  
  
^ParentChild(2)="Sharon^46647"  
^ParentChild(2,"Hobbies",1)="Yoga"  
^ParentChild(2,"Hobbies",2)="Scrap booking"  
  
^ParentChild(3)="Kaitlin^56009"  
^ParentChild(3,"Hobbies",1)="Lighting Design"  
^ParentChild(3,"Hobbies",2)="pets"
```

```
^ParentChild(4)="Melissa^56894"  
^ParentChild(4,"Hobbies",1)="Marching Band"  
^ParentChild(4,"Hobbies",2)="Pep Band"  
^ParentChild(4,"Hobbies",3)="Concert Band"  
  
^ParentChild(5)="Robin^57079"  
^ParentChild(5,"Hobbies",1)="Baking"  
^ParentChild(5,"Hobbies",2)="Reading"  
  
^ParentChild(6)="Kieran^58210"  
^ParentChild(6,"Hobbies",1)="SUBA"  
^ParentChild(6,"Hobbies",2)="Marching Band"  
^ParentChild(6,"Hobbies",3)="Rock Climbing"  
^ParentChild(6,"Hobbies",4)="Ice Climbing"
```

Paso 1:

Para esta global hay 2 conjuntos de datos repetidos, el primero se encuentra en el primer nivel de subíndices y contiene la información personal:

```
^ParentChild(1)="Brendan^45956"  
^ParentChild(2)="Sharon^46647"  
^ParentChild(3)="Kaitlin^56009"  
^ParentChild(4)="Melissa^56894"  
^ParentChild(5)="Robin^57079"  
^ParentChild(6)="Kieran^58210"
```

En el segundo está en el tercer nivel de subíndices y es una lista de los pasatiempos:

```
^ParentChild(1,"Hobbies",1)="Pit Crew"  
^ParentChild(1,"Hobbies",2)="Kayaking"  
^ParentChild(1,"Hobbies",3)="Skiing"  
^ParentChild(2,"Hobbies",1)="Yoga"  
^ParentChild(2,"Hobbies",2)="Scrap booking"  
...
```

Entonces definiremos 2 clases, una para cada bloque de datos repetidos.

Paso 2:

Tener dos clases significa que necesitamos 2 identificadores únicos, uno para cada tabla: `Example3Parent` y `Example3Child`. Es sencillo hacer esto para la tabla del padre, ya que únicamente tiene un subíndice. Para la tabla de los hijos una clave compuesta será la que utilice el primer subíndice (la referencia hacia los Padres) y el tercer subíndice (las subcategorías de los hijos).

Paso 3:

Al observar los datos es sencillo identificar 5 propiedades diferentes. Para `Example3Parent` tendremos 3 propiedades: `ParentId`, `Name` y `DateOfBirth`. Para `Example3Child` tendremos 2 propiedades: `ChildId` y `Hobby`. Aunque también necesitamos definir 2 propiedades más para este ejemplo. Estas se llaman "Relationship Properties (Propiedades de relación)". En cada clase habrá una de estas propiedades para definir la relación que tiene con la otra clase. Una clase para los padres puede tener muchas clases para los hijos, pero una clase para los hijos solo puede tener una relación con alguno de los padres.

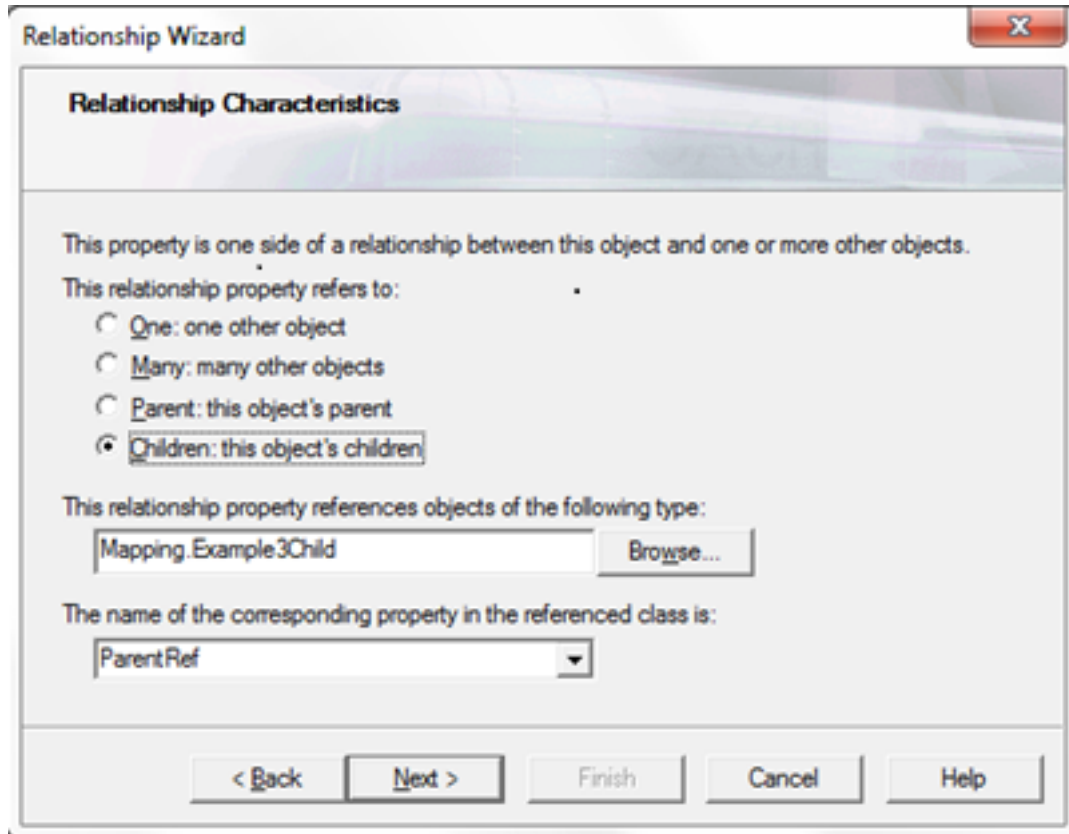
En la clase de los padres tenemos:

```
Relationship HobbyRef As Mapping.Example3Child [ Cardinality = children,  
Inverse = ParentRef ];
```

Y en la clase de los hijos tenemos:

```
Relationship ParentRef As Mapping.Example3Parent [ Cardinality = parent,  
Inverse = HobbyRef ];
```

Puede definir estas propiedades con la ayuda del "Property Wizard (Asistente para las propiedades)". Al hacer clic en el botón "Relationship radio (Proporciones de la relación)", obtendrá esta página:



Paso 4:

Muy bien, iré rápidamente hasta el paso tres donde les mostré las Propiedades de la relación. Aquí están las otras.

Example3Parent:

```
Property Name As %String;  
Property DateOfBirth As %Date;  
Property ParentId As %Integer;
```

Example3Child:

```
Property Hobby As %String;  
Property ChildId As %Integer;
```

Paso 5:

En Example3Parent obtener el IdKey es fácil, solo es un subíndice del nivel 1:

```
Index Master On ParentId [ IdKey ];
```

Para Example3Child el IdKey se basa en el subíndice 1 y el subíndice 3, pero para una relación padre e hijo obtenerlos puede ser un poco difícil. En el índice IdKey se puede hacer una lista tanto para ParentRef como para ChildId, o solamente para ChildId, ambos se compilarán correctamente.

```
Index Master On (ParentRef, ChildId) [ IdKey ];  
O  
Index Master On ChildId [ IdKey ];
```

Personalmente creo que deberíamos hacerlo de la primera manera, pero soy un poco perezoso, así que es muy probable que vea la segunda en mis ejemplos.

Paso 6:

No hay nada especial en el mapeo para Example3Parent, así que no lo revisaré a detalle. Example3Child es un poco diferente en la forma de hacer los subíndices, así que seguiré los pasos para obtener esa clase.

Paso 6a:

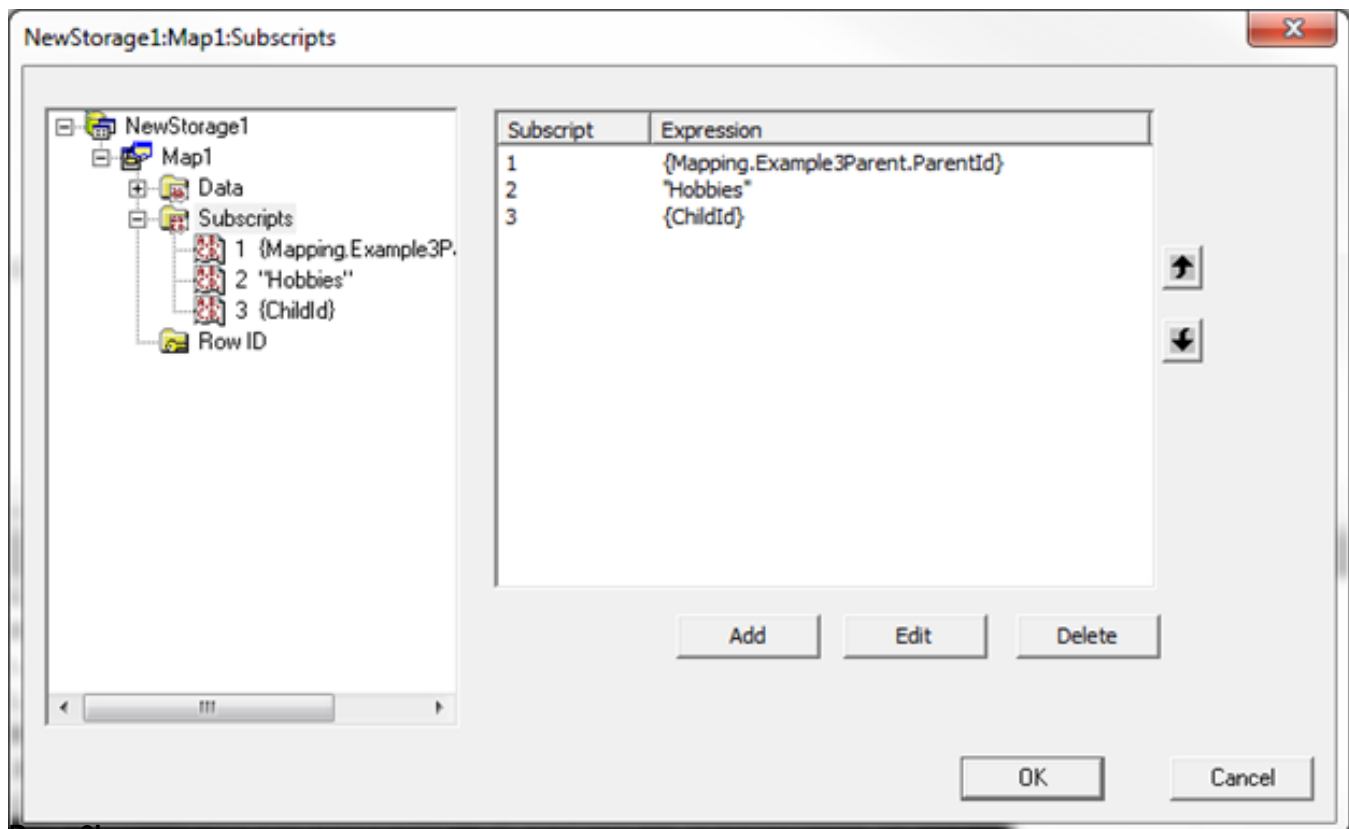
La global para la clase de los hijos se ve de esta manera:

```
^ParentChild({ParentRef}, "Hobbies", {ChildId})=""
```

Así que necesitamos 3 subíndices. El primero es el que más problemas causa a las personas. No debe utilizar la propiedad de la relación como mostré arriba, sino que necesita utilizar la propiedad de la clase de los padres. Para ello se utiliza la sintaxis del campo completo: {SchemaName.TableName.FieldName}. En este ejemplo el primer subíndice será {Mapping.Example3Parent.ParentId}.

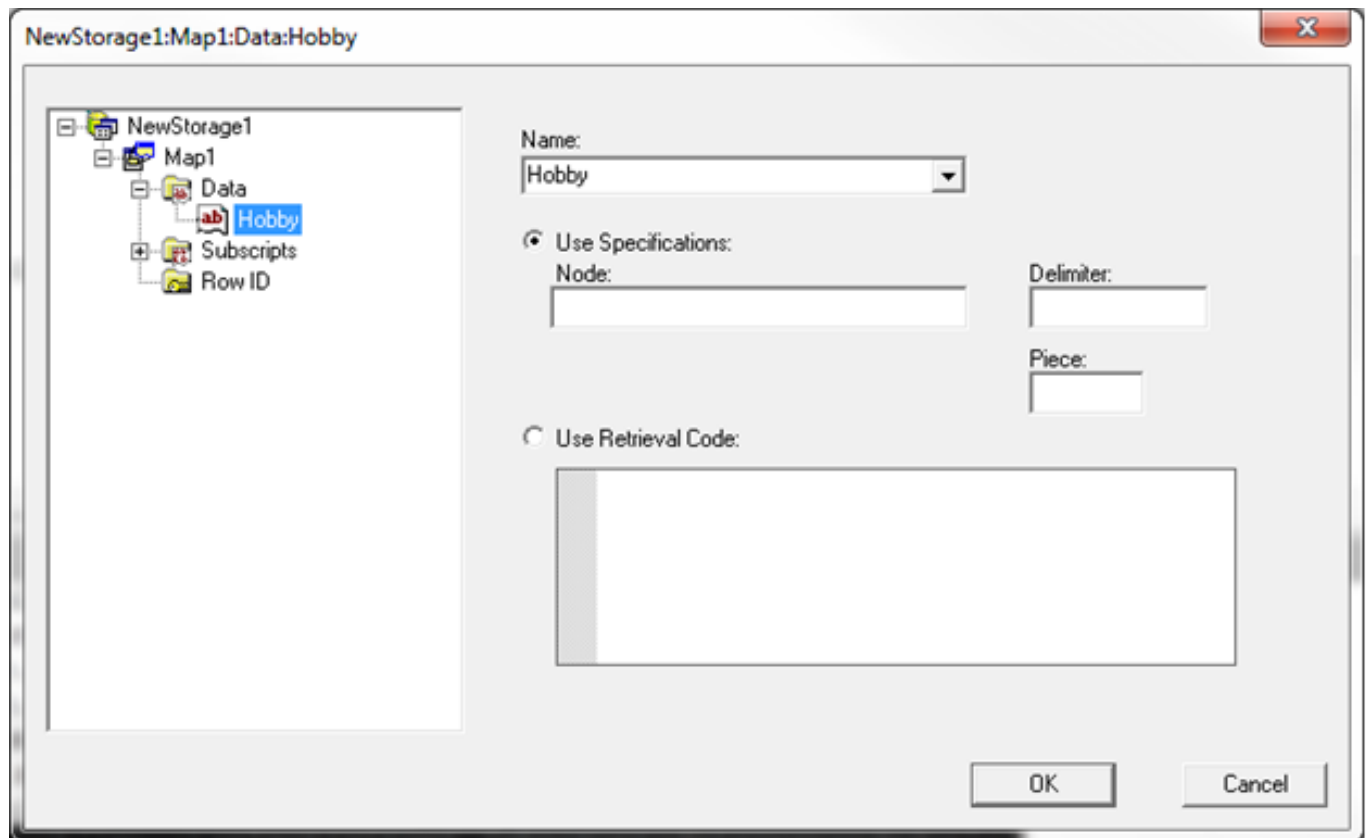
El subíndice 2 es la constante "Hobbies".

El subíndice 3 es el campo {ChildId}.



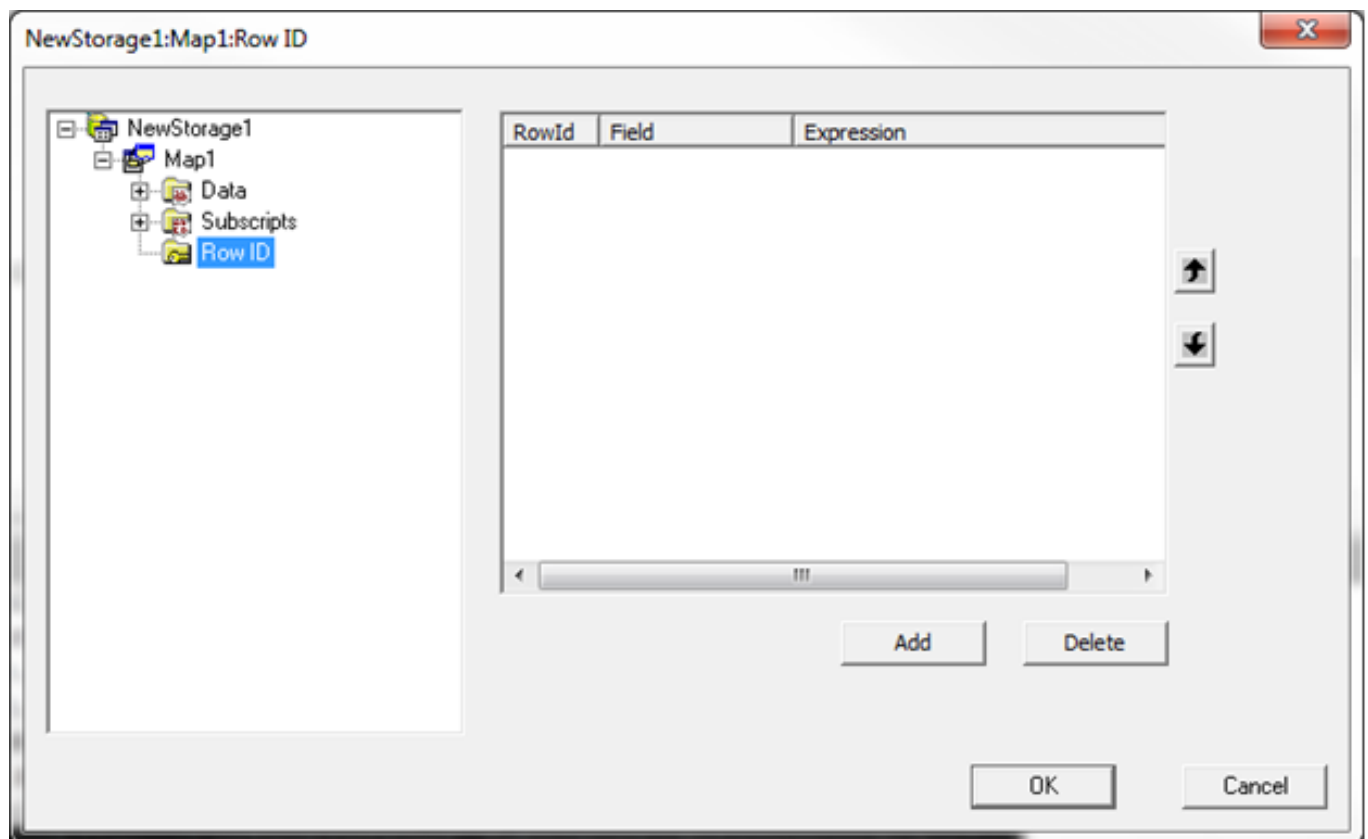
Paso 6b.

Lo que debe hacer en la sección de los datos es muy sencillo, solo se utiliza el campo {Hobby}, sin Piece ni Delimiter.



Paso 6c:

Ahora digan todos juntos, "No hay nada que ver aquí, solo es un espacio en blanco".



Paso 7:

Lo único que falta por hacer es compilarlo:

```
Compilation started on 11/28/2016 08:26:31 with qualifiers 'fuk/importselectivity=1
/checkuptodate=expandedonly'
Compiling 2 classes, using 2 worker jobs
Compiling class Mapping.Example3Parent
Compiling class Mapping.Example3Child
Compiling table Mapping.Example3Parent
Compiling table Mapping.Example3Child
Compiling routine Mapping.Example3Parent.1
Compiling routine Mapping.Example3Child.1
Compilation finished successfully in 0.900s.
```

y hacer un simple "Join" para asegurarnos de que los datos son los correctos:

```
SELECT P.ParentId, P.Name, P.DateOfBirth, C.ID, C.Hobby
FROM Mapping.Example3Parent P
JOIN Mapping.Example3Child C ON P.ParentId = C.ParentRef
```

ParentId	Name	DateOfBirth	ID	Hobby
1	Brendan	10/28/1966	1 1	Pit Crew
1	Brendan	10/28/1966	1 2	Kayaking
1	Brendan	10/28/1966	1 3	Skiing
2	Sharon	09/18/1968	2 1	Yoga
2	Sharon	09/18/1968	2 2	Scrap booking
3	Kaitlin	05/07/1994	3 1	Lighting Design
3	Kaitlin	05/07/1994	3 2	pets
4	Melissa	10/08/1996	4 1	Marching Band
4	Melissa	10/08/1996	4 2	Pep Band
4	Melissa	10/08/1996	4 3	Concert Band
5	Robin	04/11/1997	5 1	Baking
5	Robin	04/11/1997	5 2	Reading
6	Kieran	05/16/2000	6 1	SUBA
6	Kieran	05/16/2000	6 2	Marching Band
6	Kieran	05/16/2000	6 3	Rock Climbing
6	Kieran	05/16/2000	6 4	Ice Climbing

El arte de mapear Globals para Clases (3 de 3)

Published on InterSystems Developer Community (<https://community.intersystems.com>)

Tenga en cuenta que hay un campo llamado ID en la clase de los hijos (hay un campo / propiedad llamada ID en cada clase) que se compone por ParentId_"|||"_ChildId.

Para aquellos que no quieran escribir, aquí tienen un archivo con las globales y la clase: [MappingExample3.zip](#)

[#Globals](#) [#Mapeo](#) [#SQL](#) [#Caché](#)

00 1 0 0 93

Mensajes relacionados

- [El arte de mapear Globals para Clases \(2 de 3\)](#)
- [El arte de mapear Globals para Clases \(3 de 3\)](#)
- [El arte de mapear Globals para Clases \(4 de 3\)](#)

Log in or sign up to continue

Añade la respuesta

URL de fuente: <https://es.community.intersystems.com/post/el-arte-de-mapear-globals-para-clases-3-de-3>