

## 記事

[Toshihiko Minamoto](#) · 2021年2月16日 8m read

# グローバルをクラスにマッピングする技術 1/3

グローバルをクラスにマッピングする技術 (1/3)

グローバルをクラスにマッピングする技術 (1/3)

古くなった MUMPS アプリケーションに新たな生命を吹き込みたいとお考えですか？

以下にご紹介するステップを実行すれば、既存のグローバルをクラスにマッピングし、美しいデータを Object や SQL に公開できます。

本記事を含む合計 3 回の連載を通じてご紹介する簡単なステップを使えば、すべてのグローバル (特殊なものは除く) を Caché のクラスにマッピングできるようになります。

特殊なものについては、私が長年に渡って集めた様々な種類のマッピングを zip

ファイルにまとめて提供いたします。これは新しいデータを対象としたステップではありません。グローバルがないという方は、デフォルトのストレージをお使いください。

グローバルデータについて理解できないという方は、[Support@InterSystems.com](mailto:Support@InterSystems.com)

までデータをお送りください。喜んでサポートさせていただきます。

グローバルをクラスにマッピングするステップ。

1. グローバルデータが繰り返し使用されるパターンを特定する。
2. 固有キーの構成を特定する。
3. プロパティとそれぞれの型を特定する。
4. クラス内のプロパティを定義する (変数の添え字をお忘れなく)。
5. IdKey のインデックスを定義する。
6. Storage Definition を以下の手順で定義する。
  - a. 添え字を IdKey まで (IdKey を含む) 定義する。
  - b. Data セクションを定義する。
  - c. Row ID セクションは触らない。デフォルトが 99% の割合で適切なので、これはシステムに任せます。
7. クラス / テーブルをコンパイルし、テストします。

それでは例をご紹介します！

以下のようなグローバルが 2 種類 (mapping と index) あるとしましょう。

```
^mapping("Simple",1)="Brendan Bannon^55192^SQL Manager"
```

```
^mapping("Simple",2)="Nicole Aaron^63375^Support Specialist"
```

```
^mapping("Simple",3)="Kyle Baxter^61155^Senior Support Specialist"
```

```
^mapping("Simple",4)="Prasad Kari^58471^Support Specialist"
```

```
^mapping("Simple",5)="Clive Morgan^57982^Senior Support Specialist"
```

```
^index("Simple","HireDate",55192,1)=""
```

```
^index("Simple","HireDate",57982,5)=""
```

```
^index("Simple","HireDate",58471,4)=""
```

```
^index("Simple","HireDate",61155,3)=""
```

```
^index("Simple","HireDate",63375,2)=""
```

それでは、以下 7 つのステップを実行し、SQL や Object を使ってこのデータの確認ができるクラスを作成しましょう。

### ステップ 1

^mapping グローバルはいたってシンプルです。各ノードには同じ型のデータが含まれています。^index グローバルについては、^mapping グローバルの後に解説します。

### ステップ 2

最初の添え字は単なる定数です。2 つ目の添え字は、1 つずつ増加するカウンタで、データの各行で異なる数字になっています。

### ステップ 3

プロパティは、それぞれ Name、HireDate、Title のようです。55192 というのは、^index グローバルを参照したり、データの用途を把握している人に確認したりしないと、それが日付であることは分かりにくいと思います。採用日であることはなおさら分からないでしょう。2 つ目の添え字の値のプロパティを忘れずに定義しておきましょう。

### ステップ 4

```
Property Name As %String;
```

```
Property HireDate As %Date;
```

```
Property Title As %String;
```

```
Property Sub2 As %Integer;
```

### ステップ 5

Caché SQL Storage を使用する各クラスの IdKey インデックスをそれぞれ定義する必要があります。

```
Index Master On Sub2 [ IdKey ];
```

インデックスマップのインデックスを定義する必要はありませんが、定義しておくとも良いでしょう。

```
Index hireDateindex On HireDate;
```

### ステップ 6

では、ここでクラスとグローバルの間でマッピングを作成する必要があります。私はウィザードを使うのが好きなので、そのスクリーンショットをお見せします。XML を書くのが得意な方は、Caché SQL Storage を手動で定義していただけます。

[ここ](#) をクリックします。Storage Definition

には好きな名前を付けてください。私はデフォルト名を使いました。Caché SQL Storage をクリックし、「完了」をクリックします。

(補足)

### Caché SQL Storage

を作成しますと、以下のように右側のインスペクタ画面にそのプロパティが表示されますので、「SQL storage map」欄をクリックして右にある「...」をクリックします。

私は下のウィンドウで Map Name を変更しましたが、Map1 を使っても構いません。唯一入力する必要があるのは、Global Name です。「^」を付けるのを忘れないでください。永続クラスにはそれぞれデータマップが 1 つずつ必要です。複数のインデックスマップを持たせることができます。

「OK」をクリックするとウィザードを終了してしまうので、すべて完了するまではクリックしないでください。

#### ステップ 6a

次は添え字を定義したいと思います。ここでは、グローバルの添え字を IdKey まで (IdKey を含む) すべて定義します (IdKey には複数の添え字が使われる場合があります)。式ボックスでフィールドを参照するには、そのフィールドを {} の中に入れます。有効な COS 式であれば、どのような式でも入力できます。ここではシンプルに、Subscript 1 には定数を、Subscript 2 にはフィールドを使用しています。

#### ステップ 6b

データについては、IdKey の後に来るすべての添え字と「=」サインの右側にあるすべてのデータを説明しています。Node 列は、追加の添え字がある場合に使用するものです (この例は次の記事でご紹介します)。Piece と Delimiter は、グローバルの中にあるプロパティの位置を説明しています。グローバルデータを解析する場合、デフォルトでは \$PIECE コマンドが使用されます。

#### ステップ 6c

ここでは表示するものがありません。空白にしておきます。

では、^index グローバルに対してもステップ 6 を実行してください。

#### ステップ 6a

今回は添え字を 4 つ (定数 2 つ、HireDate、IdKey (Sub2)) を使います。各インデックスマップは IdKey を作成できる必要があります。通常は、添え字の中にあるものですが、データの一部となっている場合もあります。

#### ステップ 6b

追加の添え字もデータもないので、このステップは空白にしておきます

#### ステップ 6c

ここも特になしです。

#### ステップ 7

残るはクラスをコンパイルしてから、テーブルに対しクエリを実行し、データが正しく表示されることを確認するだけです。

Compilation started on 08/15/2016 15:20:58 with qualifiers 'fck /checkuptodate=expandedonly'  
Compiling class Mapping.Example1  
Compiling table Mapping.Example1  
Compiling routine Mapping.Example1.1

Compilation finished successfully in 0.270s.

```
SELECT Sub2, Name, HireDate, Title FROM Mapping.Example1
```

```
Sub2 Name HireDate Title
```

```
1 Brendan Bannon 1992-02-10 SQL Manager
```

```
2 Nicole Aaron 2014-07-07 Support Specialist
```

```
3 Kyle Baxter 2008-06-08 Senior Support Specialist
```

```
4 Prasad Kari 2001-02-01 Support Specialist
```

```
5 Clive Morgan 1999-10-01 Senior Support Specialist
```

また、自分で入力したいという方のために、XML で記述された Storage の定義を下にお見せしておきます。

/// アクティブな Storage Definition は各クラスごとに 1 つしか記述できません。

/// 各 Storage Definition にはそれぞれ複数のマップを持たせることができます。

```
Storage NewStorage1
```

```
{
```

```
<SQLMap name="HireDateIndex">
<ConditionalWithHostVars></ConditionalWithHostVars>
<Global>^index</Global>
<Subscript name="1">
<Expression>"Simple"</Expression>
</Subscript>
<Subscript name="2">
<Expression>"HireDate"</Expression>
</Subscript>
<Subscript name="3">
<Expression>{HireDate}</Expression>
</Subscript>
<Subscript name="4">
<Expression>{Sub2}</Expression>
</Subscript>
<Type>index</Type>
</SQLMap>
<SQLMap name="SimpleDataMap">
<Data name="HireDate">
<Delimiter>"^"</Delimiter>
<Piece>2</Piece>
</Data>
<Data name="Name">
<Delimiter>"^"</Delimiter>
<Piece>1</Piece>
</Data>
<Data name="Title">
<Delimiter>"^"</Delimiter>
<Piece>3</Piece>
</Data>
<Global>^mapping</Global>
<Subscript name="1">
```

```
<Expression>"Simple"</Expression>
</Subscript>
<Subscript name="2">
<Expression>{Sub2}</Expression>
</Subscript>
<Type>data</Type>
</SQLMap>
<StreamLocation>^Mapping.Example1S</StreamLocation>
<Type>%CacheSQLStorage</Type>
}
```

自分で入力したくないという方は、グローバルとクラスが記述されたこちらのファイルをお使いください:  
[mappingexample1.zip](#)

さらに知識を深めたいという方は、是非[パート2](#) もお読みください

[#SQL](#) [#グローバル](#) [#マッピング](#) [#Caché](#)

---

### ソースURL:

<https://jp.community.intersystems.com/post/%E3%82%B0%E3%83%AD%E3%83%BC%E3%83%90%E3%83%AB%E3%82%92%E3%82%AF%E3%83%A9%E3%82%B9%E3%81%AB%E3%83%9E%E3%83%83%E3%83%94%E3%83%B3%E3%82%B0%E3%81%99%E3%82%8B%E6%8A%80%E8%A1%93-13>