
Artículo

[Kurro Lopez](#) · 16 abr, 2019 Lectura de 3 min

Copiar parte de las propiedades de un objeto a otro

A veces, necesitamos copiar parte de las propiedades de un objeto en otro diferente.

La forma mas facil sería hacer lo siguiente:

```
Set obj1.FirstName = obj2.FirstName
```

```
Set obj1.SecondName = obj2.SecondName
```

¿Que sucede si el objeto contiene un gran número de propiedades? ¿Y si solo necesitamos extraer un importante grupo de datos, y complementar la información en otro objeto?

Tenemos las siguientes clases:

```
Class Kurro.PersonalInfo Extends %Library.SerialObject
{
    /// Code of personal
    Property Code As %String;

    /// First name of personal
    Property FirstName As %String;

    /// Second name of personal
    Property SecondName As %String;

    /// Date of birthday of personal
    Property DateOfBirth As %DateTime;

    /// Passport ID number
    Property PassportId As %String;
}
```

```
Class Kurro.NameInfo Extends %Library.SerialObject
{
    /// First name of personal
    Property FirstName As %String;

    /// Second name of personal
    Property SecondName As %String;

    /// Date of birthday of personal
    Property DateOfBirth As %DateTime;
}
```

```

/// Relationship
Property Relationship As %String;

}

```

Relleno el objeto PersonalInfo

```

ENSDEMO>zw personal
personal=<OBJECT REFERENCE>[1@Kurro.PersonalInfo]
+----- general information -----
|      oref value: 1
|      class name: Kurro.PersonalInfo
| reference count: 2
+----- attribute values -----
|      Code = "FLH01"
|      DateOfBirthday = 47878
|      FirstName = "Francisco"
|      PassportId = "ESP-123456"
|      SecondName = "Lopez"
+-----

```

Creo la siguiente clase de utilidades

```

Class Kurro.Utills.Functions
{

/// Copy properties
Method CopyProps(Output obj) As %Status [ CodeMode = objectgenerator ]
{
    set properties = %compiledclass.Properties
    for i=1:1:properties.Count()
    {
        #dim prop As %Dictionary.CompiledProperty = properties.GetAt(i)
        if prop.Name'="",prop.Name'["%"
        {
            set propName = prop.Name
            if prop.Name["_"]
            {
                set propName = ""_"_prop.Name_"
            }
            do %code.WriteLine(" If (obj.ExistsProperty(""_"_propName_"")) { set obj.
            "_propName _"=.."_propName_" } ")
        }
    }
    quit $$$OK
}

/// Check if a property exists
ClassMethod ExistsProperty(pPropName As %String = "") As %Boolean
{
    Set tExists = 0
    If (pPropName '= "")
    {
        Set tExists = ##class(%Dictionary.CompiledProperty).IDKEYExists($classname($t

```

```
his),pPropName)
    }
    Quit tExists
}

}
```

Ahora, necesitamos extender también nuestras clases a la clase de utilidades.

```
Class Kurro.PersonalInfo Extends (%Library.SerialObject, Kurro.Utills.Functions)
Class Kurro.NameInfo Extends (%Library.SerialObject, Kurro.Utills.Functions)
```

Probar como funciona usando el siguiente código:

```
set personal = ##class(Kurro.PersonalInfo).%New()
set personal.Code="FLH01"
set personal.FirstName="Francisco"
set personal.SecondName="Lopez"
set personal.DateOfBirthday=$ZDATEH("1972-02-01",3)
set personal.PassportId ="ESP-123456"
set name = ##class(Kurro.NameInfo).%New()
do personal.CopyProps(.name)

zw name
```

Este es el resultado:

```
ENSDEMO>zw name
name=<OBJECT REFERENCE>[3@Kurro.NameInfo]
+----- general information -----
|      oref value: 3
|      class name: Kurro.NameInfo
| reference count: 2
+----- attribute values -----
|      DateOfBirthday = 47878
|      FirstName = "Francisco"
|      Relationship = ""
|      SecondName = "Lopez"
+-----
```

Importante: Solo se podrá copiar el contenido de una clase que extienda de esta funcion, porque necesita el método ExistsProperty.

Espero que este truco te sea tan util como me ha sido a mi.

Un Saludo

[#Consejos y trucos](#) [#ObjectScript](#) [#Principiante](#) [#Caché](#)