
Question

[Rochdi Badis](#) · Dec 14, 2022

Property definition principles

Hi Guys,

I'm trying to get head around this principle of instance variables and the advantage of it.

I found this principle used a lot by my predecessor in some properties definition and I'm wondering why don't we just use the property as simple as is, its creating two properties sCtg that contain the value and Ctg is a calculate to get the value of sCtg, is there advantage of fast accessing or... !?

and also the use of DISPLAYLIST & VALUelist does this brings any advantage vs defining a standard property (eg.fast access!), so instead of have to do Valuetist "H" and Dispay "Hot" why just a standard property as string containing "Hot"?

```
Property Ctg As %Text(MAXLEN = 256
) [ Calculated, SqlComputeCode = { Set {Ctg}=##class
(MSDS.Serenity.Kit).GetKitCtg({ID})}, SqlComputed ];
```

```
Property sCtg As %Text(MAXLEN = 256);
```

```
ClassMethod GetKitCtg(kitId As %String) As %String
{
    set oCK = ..%OpenId( kitId)
    quit:'$isObject( oCK) ""
    quit oCK.Ctg
}
```

```
Method CtgGet() As %Text
{
    quit i%sCtg
}
```

```
Method CtgSet(name As %Text) As %Status
{
    set i%sCtg = name
    quit $$$OK
}
```

Thanks

[#Ensemble](#)

Product version: Ensemble 2014.1